Software Engineering Institute

Adaptive Flow Control for Enabling Quality of Service in Tactical Ad Hoc Wireless Networks

Jeffrey Hansen
Scott Hissam
B. Craig Meyers
Ed Morris
Daniel Plakosh
Soumya Simanta
Lutz Wrage

December 2010

TECHNICAL REPORT
CMU/SEI-2010-TR-030
ESC-TR-2010-030

Research, Technology, and System Solutions Program

http://www.sei.cmu.edu

Carnegie Mellon

This report was prepared for the

SEI Administrative Agent
ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2100

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

For information about SEI publications, please visit the library on the SEI website (www.sei.cmu.edu/library).

Table of Contents

List of Figures

List of Tables

Acknowledgments

We sincerely acknowledge the support given us by the Software Engineering Institute management during the course of this research. We thank Linda Northrop for establishing the laboratory facilities that provided the space and equipment necessary to conduct our research and experiments over the long term. We thank Alexander Bordetsky, Associate Professor, Ray Buettner, Associate Professor, Mike Clement, Research Associate, and Marianna Verett, Research Associate—all from the Naval Postgraduate School (NPS)—for their resolute support of the experiments we conducted during the USSOCOM-NPS Capabilities Based Experimentation held at NPS's Tactical Network Test bed at Camp Roberts, CA.

We also thank Paul Clements, Mark Klein, Gabriel Moreno, Linda Northrop, and Patrick Place for their insightful and thorough review. We extend that word of thanks to John Morley, who had the distinct pleasure of editing this report and did so in a surprisingly short amount of time.

Executive Summary

Many visions for the future involve pervasive computing technology that links people and devices together to solve complex problems. The Global Information Grid (GIG), for example, as well as the TeraGrid and the Smart Grid, are large-scale endeavors in which computing resources are increasingly interconnected by wired, high bandwidth networks.

However, in more localized crisis or tactical scenarios, the network infrastructure for users such as emergency responders or warfighters is

- wireless
- often assembled *on the spot* without a preexisting infrastructure (i.e., an ad hoc network)
- subject to changing topology as individual nodes enter, leave, or move in the environment

These users are frequently faced with an impoverished network computing environment that

- has lower bandwidth than wired or preplanned wireless networks
- experiences rapid changes in available bandwidth due to its ad hoc nature
- encounters network interruptions due to interference from buildings and terrain, environmental conditions, and jamming (in warfighting)

Further, the amount of data on these networks can place a crushing load on network resources—increasing latency and data loss. In effect, users and assets (e.g., various soldiers on combat patrol, command and control systems, data feeds from UAVs) are competing for very limited network resources. Estimates made by the Congressional Budget Office state that bandwidth for the U.S. Army was expected to fall short of peak demand by a factor of 10 [CBO 2003].

When sufficient bandwidth is not available, applications in the field may simply stall as the network becomes oversubscribed. In many tactical military situations, stalls become catastrophic and force soldiers to make decisions based on inadequate or out-of-date information. A key to mission success for individual users, as well as to mission success overall, is therefore the way that the ad hoc wireless network and its associated resources are managed. The goal of managing the network and resources is to ensure that users and assets with the most critical needs gain access to those resources (where criticality is determined by a wide range of factors).

Several strategies to manage limited resources in ad hoc wireless networks have been developed in research and practice. One common strategy involves developing a simple allocation scheme that allocates network resources using statically assigned priorities. In many cases, however, these simple prioritization schemes are not sufficient because they do not necessarily align with the needs of the mission. When the supply of bandwidth becomes inadequate during combat, military operations officers sometimes enforce a crude priority scheme by literally “pulling the plug” on radio equipment and computers to free up bandwidth for high-priority messages [Wilson 2005]. This strategy and other simple priority schemes have the unfortunate side effect of prematurely starving lower priority data transmissions. Also, because of the variable nature of ad hoc wireless networks (e.g., variable bandwidth based on aspects such as environmental factors and antenna position), it is not possible to guarantee delivery of even high-priority transmissions.

Alternative approaches involve strategies that allow both the network and applications to adapt to conditions and thus maintain a level of service that, while not completely adequate, is sufficient for some degree of continued functioning. In these approaches, an application can adopt a strategy that requires less bandwidth on receiving notice of inadequate bandwidth availability. Low-priority transmissions, thus, avoid starvation and continue to progress, although at a lessened pace or with reduced capability. But many of these alternative approaches rely on the ability to know, or make sophisticated estimates or predictions of network capacity—a notion impractical in a mobile, ad hoc, wireless networks whose topology is dynamically changing.

The approach introduced in this report, Adaptive Quality of Service (AQoS), builds on and integrates several of these alternative techniques and ideas, including the classification of flows, the prioritization of flows, and network congestion feedback to applications. Furthermore, rather than trying to continually estimate or predict available bandwidth, AQoS can monitor queues deployed in the networking infrastructure as an *early warning system* for network congestion. These observations are used by the network infrastructure to notify mission-critical applications of such conditions. Applications, where the consequences of adaptation are best understood, can use this information to adapt to changing network service levels and continue to progress toward achieving mission goals.

This report documents the results from 18 experiments that were conducted in the field and laboratory to investigate AQoS. In these experiments, the networking infrastructure reacted to, and mission applications adapted to, dynamic changes in available network capacity in both undersubscribed and oversubscribed (or overloaded) scenarios.

In the undersubscription network scenarios, the demonstrated performance of AQoS was found to be no worse than that of the other traffic management policies defined for our investigation. More importantly, for oversubscription scenarios, AQoS's approach showed that applications were able to adapt, in a controlled manner, their bandwidth demand to reductions in available network bandwidth thereby avoiding uncontrolled packet loss due to congestion.

Although we limited our investigation to a specific situation, our assessment of the AQoS approach is encouraging. Achieving continued and usable service for high-priority applications in the face of dynamically changing and diminishing (or conversely, increasing) network capacity without the need to reserve network bandwidth or to know, *a priori*, available network bandwidth is an important step. In addition, we developed model applications for our experiments as a means to understand longer term questions such as "Can applications be built that provide requisite flexibility for edge users, yet make efficient use of ad hoc wireless network resources?"

In future work, we intend to address many of the limiting factors of our experiments to date. Specifically, we will investigate utility-based degradation of service in response to resource shortfalls. In our experiments, the low-priority task could be completely starved in order to meet the full demands of the high-priority task. Under a utility-based approach, the notion of priority is generalized so that a low-priority task could be allowed to operate at a minimal level in exchange for a slight degradation of a high-priority task. These allocation decisions lead to increased system-wide benefit. Further, the wireless network in our experiment had only two nodes. Ultimately, AQoS will have to operate in multi-node, wireless networks such as mobile ad hoc wireless mesh networks that are self-configuring and self-healing.

Abstract

Wireless networks for emergency responders and military personnel operating in tactical situations are often assembled without any preexisting infrastructure (i.e., ad hoc) and are subject to changing topology as nodes enter or leave service or move (i.e., are mobile) in the environment. These networks often have lower-than-optimal bandwidth and can experience further bandwidth reductions due to disadvantageous topologies and other factors. In addition, needed applications must compete for possibly diminishing bandwidth. As a result, such networks are frequently oversubscribed: they cannot fully meet the quality of service (QoS) expectations of all applications.

This report provides an overview of approaches for satisfying QoS expectations in ad hoc wireless networks assembled to support high-criticality crisis and tactical scenarios. It illustrates that these approaches are adaptations of approaches used in wired (often fixed) infrastructures where bandwidth is known and interference is not the norm. It documents and provides experimental evidence for the Adaptive QoS (AQoS) approach that allows applications to adapt bandwidth demand to wireless network conditions without the need to know, estimate, or predict available bandwidth. AQoS informs applications that network oversubscription is occurring, thereby allowing them to continue to operate, albeit at diminished rate or capacity, to meet mission needs.

1 Introduction

Many visions for the future involve pervasive computing technology that links people and devices together to solve complex problems. This is a goal sought by the

- **Global Information Grid (GIG),** which aims to provide globally interconnected information capabilities and processes for warfighters
- **TeraGrid,** which integrates high performance computers and networks, data, and tools in support of scientific research
- **Smart Grid,** which manages the electric power grid through computer-supported interaction between energy producers, consumers (e.g., homes, business, industries) and the power grid itself

Significant progress is being made in all these endeavors as computing resources are increasingly interconnected by wired, high-bandwidth networks.

The problems facing users in crisis and tactical scenarios are different, however. The networks typically accessed by these users are wireless, often assembled on the spot without preexisting infrastructure (i.e., an ad hoc network), and subject to changing topology as nodes enter service, leave service, or move in the environment [Meissner 2002]. The user, whether a soldier on patrol or an emergency responder, is frequently faced with impoverished network computing environments that

- have lower bandwidth than wired or preplanned wireless networks
- are subject to rapid changes in available bandwidth due to factors such as the number of nodes on the network, the network topology, and the orientation of antennas
- are subject to network interruptions due to interference from buildings and terrain, environmental conditions, and (in the case of warfighting) jamming

These problems are particularly acute for a modern military that is reliant on wireless, ad hoc networks to provide deployment flexibility without pre-positioned network infrastructure.

Current tactical networks employed by the military are frequently overloaded and the demands placed on them are increasing. A modern soldier connected to an ad hoc wireless network can access map data; information about specific locations, buildings, or persons; and video data from an unmanned aerial vehicle (UAV). In the very near future, a soldier will have access to information from multiple back-end databases and a range of sensors, including optical and infrared cameras, detectors for chemical and biological threats, and devices that detect movement. In addition, the soldier will transmit increasing volumes of information through the ad hoc wireless networks back up the chain of command.

All this data can place a crushing load on network resources—diminishing bandwidth, increasing latency, or reducing reliability. In effect, users (e.g., various soldiers on combat patrol, command and control systems, data feeds from UAVs) are competing for these very limited network resources. In fact, previous estimates made by the Congressional Budget Office state that bandwidth for the U.S. Army was expected to fall short of peak demand by a factor of 10 [CBO 2003].

When sufficient bandwidth is not available, the user's application may simply stall. In the case of many tactical military situations, stalls become catastrophic and force soldiers to make decisions based on inadequate or out-of-date information. A key to mission success for individual users, as well as to mission success overall, is therefore the way that the ad hoc wireless network and its associated resources are managed in overloaded conditions. The goal of managing the network and resources is to ensure that users with the most critical needs gain access to those resources (where criticality is determined by a wide range of factors).

Several strategies have been developed to manage limited resources, such as bandwidth, in ad hoc wireless networks. One common strategy involves a simple allocation scheme that assigns network resources based on highest priority. For example, one scheme assigns priority, and therefore a *claim* on network resources, first to network management activities, followed by (in decreasing priority) to VoIP, UDP, and TCP data streams.[1] Thus, when adequate bandwidth is not available to satisfy all requests, users receiving data over TCP will stall first, followed by other data streams. It is easy to imagine other simple priority schemes based on factors such as the source and type of the request.

In some cases, however, these simple schemes are not sufficient because they do not consider the context in which user applications operate. Simple priority schemes also have the unfortunate side effect of starving lower priority data transmissions. And, because of the variable nature of ad hoc wireless networks (e.g., variable bandwidth based on environmental factors, antenna position, and the like), it is not possible to guarantee delivery of even high-priority transmissions. In cases like this, military operations officers sometimes enforce a crude priority scheme by literally "pulling the plug" on radio equipment and computers to free up bandwidth for high-priority messages [Wilson 2005].

Alternative approaches involve strategies that allow both the network and applications to adapt to conditions and thus maintain a level of service that, while not completely adequate, is sufficient for some degree of continued functioning. In these approaches, an application, on receiving notice of inadequate bandwidth, can adopt a strategy that requires less bandwidth. This allows low-priority transmissions to avoid starvation and continue to progress, although at a lessened pace due to slower data transmission or with reduced capability such as lower image resolution. Perhaps the most accessible analogy to the approach involves revelers at a Super Bowl party watching a critical play in the football game. If bandwidth to transmit a high definition image is suddenly unavailable because bandwidth is needed for a high-priority emergency elsewhere in the network, revelers are typically very unhappy (maybe angry) as the picture freezes or breaks up.[2] If, on the other hand, the satellite receiver receives notice and adapts to receive standard definition images, and the HDTV itself can be notified and similarly adapt, partiers may be slightly disappointed but they will still be able to view the action.

---

[1] VoIP is voice over internet protocol; UDP is user datagram protocol; and TCP is transmission control protocol.

[2] HDTV has a very limited ability to adapt to signal loss. If the signal drops out of the adaptation range, the HDTV is "starving" and the image will freeze or break up. Standard definition images require less bandwidth to transmit.

1.1 Our Approach: AQoS

This report describes an initial step to addressing the critical problem of managing resources, specifically bandwidth, in mobile ad hoc wireless networks common to tactical environments. Our work involves answering these and other questions:

- Can a shared ad hoc wireless network infrastructure be managed to ensure that sufficient quality of service (QoS) is maintained to achieve multiple simultaneous missions?
- What strategies can be employed to enhance the value derived from the resources that are available—particularly bandwidth—and can these strategies respond to the rapidly changing context of tactical environments?

The approach, Adaptive Quality of Service (AQoS), is motivated by a series of experiments addressing these and related questions conducted at the Carnegie Mellon® Software Engineering Institute (SEI) and at the Naval Post-Graduate School (NPS) Field Experimentation Cooperative Tactical Network Topology testbed[3] [Klein 2008, Plakosh 2008]. These past experiments, including the research described in this report, investigate the application of theories, technologies, and software engineering techniques that are practical and relevant to the future of tactical networks.

Our work differs from prior work in wireless ad hoc networking in the following ways:

- Applications are able to adapt in response to indicators of insufficient capacity of the underlying network.
- The networking infrastructure does not need knowledge of application data semantics.
- Other than providing fairly straightforward indicators of network state, there is no need for the network infrastructure to predict bandwidth capacity.

In essence, when applications place demand on a mobile ad hoc wireless network that has a limited and volatile capacity, AQoS is able to

- detect situations where capacity is outstripped by demand and inform applications of this condition
- prioritize application data traffic so that loss is incurred by applications of low(er) priority before those of higher priority

In order to continue carrying out its mission when resources are scarce, the application can adapt to congestion notification by reducing its demand. The advantage of this approach is that QoS decisions are controlled by the application, where the consequences are well-understood.

The intent for this work is to move these theories and ideas from concepts on paper to refined ones that are proven to work in real-world environments.

® Carnegie Mellon is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

3 For more information, visit http://cenetix.nps.edu/cenetix/.

1.2 Roadmap for this Report

Section 2 provides an overview of prior work on QoS problems for ad hoc wireless networks as well as related research to overcome such problems. Section 3 provides details of our approach, which includes the use of model problems to identify capabilities needed and analyze specific strategies and field experiments to validate that our strategy is viable in increasingly realistic settings. Section 4 describes our specific activities, experiments, and quantitative results. Section 5 lays out our future work intended to address current limitations of our approach. Section 6 concludes the report.

2 Background

In designing U.S. Department of Defense (DoD) computing systems—particularly those serving tactical edge users—system and software architects must consider the highly dynamic nature of the operating environment. This dynamic nature is reflected in widely varied and often rapidly changing network infrastructures and bandwidth availability and often results in sudden and unpredictable bandwidth shortfalls. Bandwidth scarcity forces applications deployed for use in these environments to compete with one another for networking and other resources to accomplish the mission.

Arbitration is the means used to determine which applications get priority, when and for how long those applications can run, and the amount of bandwidth and other resources allocated to each application. Currently, arbitration is a continuous, human-intensive process. This process is only exacerbated by the dynamic and minute-to-minute volatility of the ad hoc wireless mobile networks that tactical edge users rely on, where bandwidth, quality of signal, delay, and other characteristics are constantly changing variables.

When the dynamics of the environment and the effect on networks and systems are under appreciated or under estimated, perhaps due to unforeseen circumstances, "pulling the plug" or otherwise disabling some radios or switches is not a good solution. Such a dramatic approach may provide adequate performance for subjectively determined high-priority transmissions, but it will completely starve or otherwise disable others that may ultimately be equally or more critical to the mission. As such, mission success using ad hoc wireless networks in crisis and tactical environments may well depend on the ability to avoid such situations where users must manually intervene. Instead, we should seek to engineer solutions that embrace the dynamic nature of the tactical edge environment rather than

- ignoring the inherent volatility of the ad hoc wireless network and attempt to make the network appear as a wired network to the mission applications
- assuming that mission priorities are static, mission applications faithfully express their requirements for bandwidth and other resources, and mission and/or network operations can be effectively centralized and coordinated
- assuming mission applications can be designed independently of the consideration of the networking environment (other than priority assignments), especially when inconsistent reliability and connectivity, lost messages, and variations in available bandwidth are common in that environment

Approaches that strive to accommodate some of these issues tend to focus directly on the ad hoc wireless networking technology. Rarely, if at all, do they consider the architecture of the mission application or the inherent inability to know, *a priori*, all the information needed to reason about the network and the interaction of the mission applications at all scales.

Furthermore, the majority of approaches to QoS in ad hoc wireless networks assume that applications have rigid requirements and attempt to force-fit the network to those requirements. Typically these techniques attempt to make the network compensate for the resource shortfalls using tech-

niques such as QoS rerouting, attempting to make the wireless network behave like a wired network from the application perspective. These applications then operate under the false assumption that failures and changes in resource capacity are rare.

We next discuss techniques for management of QoS in wired networks, followed by a discussion of the state of research for management of QoS in ad hoc wireless networks.

2.1 QoS in Wired Network Infrastructure

The core computer network protocols we use today such as IP and TCP/IP were designed to provide primarily *best-effort* service where there are no guarantees of QoS, but a best effort is made to provide the service given network traffic and other constraints. In this scheme, resources are available equally to all computers sharing the network, with resources being granted on a first-come, first-service basis. As long as there is a surplus of resources available, this scheme works well, but as applications demand more network resources (eating up the surplus) and are subject to stricter timing requirements, the best effort paradigm rapidly results in unsatisfactory service for all users.

In wired networks, one of the early efforts to address the problems caused by best effort service was resource reservation protocol (RSVP) [Zhang 1993, Braden 1997, and Polk 2006]. RSVP is a transport layer protocol that can be used to make end-to-end resource (e.g., bandwidth) reservations across a network. It uses a signaling mechanism to set up a bandwidth reservation at each hop on the path from the source to the destination. RSVP does not require that every router on the end-to-end path support RSVP.[4] It is often the case that routers near the source and destination might support RSVP, while core internet routers do not. In this case, the RSVP reservations will be made only at the routers supporting RSVP with *best effort* on the routers in between. Since congestion often occurs near the source and destination of a request rather than in the usually much higher capacity core routers, use of RSVP often provides acceptable QoS.

However, RSVP is only a signaling protocol concerned primarily with relaying reservation information to nodes in the network. It does not directly enforce bandwidth reservations and so it must work in conjunction with an enforcement mechanism. One enforcement mechanism frequently used with RSVP is WFQ (Weighted Fair Queueing) [Demers 1990]. In WFQ, different flows or classes of traffic can be configured to be given a fraction of the bandwidth on a link. This can be thought of as maintaining a separate queue for each flow, with each flow being serviced in proportion to the fraction of the link bandwidth that has been allocated to it. Compared to first-in first-out (FIFO) service in a normal router, WFQ guarantees that the service offered to a flow will not exceed certain worst-case bounds in terms of bandwidth capacity.

RSVP is an example of an integrated service (IntServ) [Braden 1994]. In IntServ mechanisms, QoS is managed on a per-flow basis with each flow having the ability to specify its required level of service. This ability allows for fine-grained control over QoS, but it means that each router in the network must maintain a flow table indentifying each flow and its QoS requirements.

---

[4] A key requirement for any network, including wireless networks, is the ability to determine how to move information from one point to another. This is called routing. A router is a device that moves the information.

An alternative approach to internet QoS is Differentiated Services (DiffServ) [Nichols 1998, Blake 1998]. In the DiffServ model, mechanisms merely distinguish between a small number of different classes of traffic, rather than creating reservations for specific flows. Each packet for a flow is then tagged at the source with a class indicating its QoS needs. DiffServ provides only aggregate QoS; it does not provide the same level of guarantee that an end-to-end resource reservation would. However, DiffServ is far easier to implement and does not require that routers maintain a table of flows needing QoS management. Instead, DiffServ simply supports a small fixed-sized number of classes.

While resource reservation and enforcement are important aspects of QoS management, making decisions on which applications should receive what level of resource is a fundamental problem. One approach to addressing this problem is Q-RAM (QoS Resource Allocation Model) [Lee 1999]. In the Q-RAM approach, applications are assumed to be capable of operating at two or more levels of service across one or more QoS dimensions. For example, a video application might be able to operate at four different frame rates and five different resolutions. Users of these applications derive more satisfaction from applications running at the high levels of service and less from those running at the lower levels of service. In Q-RAM, the degree of satisfaction is quantized[5] using a value called *utility*. Utility increases as the level of service on a dimension increases. Furthermore, utility can also be weighted depending on the importance of a user or the task for which it is being used. For example, a video connection for command-and-control in an active battle situation would receive higher utility than one for a soldier talking to his family back home.

The core of the Q-RAM approach is an optimization algorithm that takes as input

- a **user profile** containing the utility assignments to the different levels of service
- an **application profile** specifying the resource requirements for each level of service
- a **resource profile** specifying the amount of each resource available

This information is then used to compute the allocation of resource that will meet resource constraints and have the highest total system utility. While this problem is in general NP-hard,[6] Q-RAM uses heuristics that can get within a fixed bound of optimal when certain conditions are met. Specifically, applications should see a diminishing rate of return on the value of resources, with each additional increment in resource receiving less utility. This restriction allows Q-RAM to employ convex optimization techniques resulting in an O(n log n) optimization heuristic.

Q-RAM has been employed as part of a QoS management system for wired networks in the Amaranth system [Hoover 2001]. In Amaranth, applications register their profiles with the session coordinator. When an instance of an application starts, it requests resources from the session coordinator. The session coordinator computes an optimal operating point for the new application instance and other application instances running in the system and sends QoS control messages to the application instances directing them to operate at a new level of service. The session coordinator attempts to leave a portion of the resource unallocated and ready to be used by future applica-

---

[5] Quantization is a procedure to constrain something from a larger to smaller set of values.

[6] NP-hard (non-deterministic polynomial-time hard) describes a class of intrinsically very hard problems. For more information, visit http://www.itl.nist.gov/div897/sqg/dads/HTML/nphard.html.

tion instances [Hansen 2001]. Leaving a small portion of the resources unallocated can dramatically decrease the number of QoS control messages that must be sent, reducing the frequency with which application QoS must be changed.

2.2 QoS in Ad Hoc Wireless Networks

Unfortunately, the nature of ad hoc wireless networks makes these approaches to wired network QoS infeasible [Chen 2004, Zhang 2005]. Resource capacity in ad hoc wireless networks is often unknown and subject to rapid changes. Furthermore, ad hoc wireless networks are generally decentralized and are subject to frequent changes in topology with no node having precise knowledge of the network. Because of these features, offering hard-guaranteed QoS (i.e., 100% assurance that target QoS is met) in an ad hoc network is generally considered impractical. Most approaches look to provide soft QoS guarantees in which QoS is allowed to occasionally fall short of the target level [Chen 2007].

Special routing protocols have been developed for use in ad hoc wireless networks. Most commonly used wireless routing protocols are concerned only with finding a route, not with the quality of the route. Two of these routing protocols are OLSR (Optimized Link State Routing Protocol) [Clauson 2003] and B.A.T.M.A.N. (Better Approach To Mobile Ad-hoc Networking) [Batman 2010]. OLSR is an IP routing protocol in which link connectivity is established by periodic HELLO messages sent by each node. Nodes listen for these messages to determine their neighbors and use additional messages to build and maintain a routing table for the network. A subset of the nodes are elected as MPRs (Multi-Point Relays) in such a way that every node can directly communicate with at least one MPR. The MPRs share information, so that each has a complete routing table. A proactive routing protocol, OLSR determines routes before they are needed. If changes to the network are relatively rare, OLSR can result in faster forwarding of messages, but at the expense of substantial overhead in maintaining the routing tables.

Unlike OLSR, the B.A.T.M.A.N. routing protocol does not maintain complete routing information at each node. Instead, each node may only know the general right direction in which to forward a packet. B.A.T.M.A.N. also differs from OLSR in the way in which it determines paths. Each node broadcasts originator messages to advertise its existence to neighbors. These messages are forwarded until all nodes are aware of the others. Since the originator messages are sent unreliably, however, poor quality links may lose the originator messages, causing the better quality links to be favored. While OLSR has been shown to deliver slightly better throughput for smaller scale networks, B.A.T.M.A.N. demonstrates better routing performance (e.g., less packet loss, faster healing time) and is better able to maintain connectivity in larger networks [Abolhasan 2009].

Unfortunately, both OLSR and B.A.T.M.A.N. were designed primarily for finding a feasible route between points in the network with minimal regard to the quality of that route [Chen 2007, Ge 2003]. Both protocols only consider whether two nodes can communicate, ignoring the quality of the link. QoS-aware protocols must consider factors that can affect the quality of links, including

- *Interference from other nodes*—Since all nodes share the media, a node may have to wait for other nodes transmitting on the same media. Due to the distributed nature of ad hoc wireless networks, it is extremely difficult to predict the amount of interference of this type since nodes do not have knowledge of traffic on other nodes.

- *Intra-flow interference*—If a flow requires multiple hops, transmissions for each hop may interfere with other hops. This may result in capacity as low as $1/7^{th}$ of that available for a single hop [Li 2001].
- *Environmental factors*—Difficult-to-define factors such as the position and orientation of the antenna, topography, atmospheric conditions can also have an unpredictable effect on wireless link quality.

In order to address these issues, researchers have been developing new strategies to provide QoS for ad hoc wireless networks. They have focused on router-level issues in the following key areas [Chen 2007]:

- **Resource Estimation:** determining the amount of resource available on a link or end-to-end
- **Route Discovery:** finding paths with low latency or sufficient bandwidth to use for a flow
- **Resource Reservation:** allocating resources to a specific flow or flows
- **Route Maintenance:** maintaining and enforcing reservations or service on a path and responding to changes in link quality
- **Route Selection:** choosing from among multiple potential routes

In the following sections, we discuss each of these issues and identify some of the approaches researchers and practitioners have taken to address them.

2.2.1 Resource Estimation

Since many approaches to QoS involve reserving resources for a task from a fixed-sized pool, determining the amount of resources available is an important issue. In a wired network, the capacity of a link is typically fixed according to the selected hardware. However, in an ad hoc wireless network the amount of resources available is very difficult to determine and is constantly changing due to intra-node and environmental interference. Not knowing the amount of available resources is of particular concern when the nodes in these networks are mobile. Wireless radios will typically change transmission modes in response to the link quality, but the transmission mode is not a reliable indicator of the amount of bandwidth available.

2.2.1.1 Link Busyness Ratio

To estimate the available bandwidth on a link, the carrier sense capability in IEEE 802.11 is often used to compute the link busyness ratio [Chen 2007, Ge 2003, Chen 2005, and Shen 2008]. This capability allows an application to measure the fraction of time a channel is busy, including transmit and receive time on the measuring node as well as transmissions between other pairs of nodes using the channel. However, the link busyness ratio is more a measure of the bandwidth used and is not a good indicator of additional capacity available.

2.2.1.2 Active Probing

Some researchers have tried to address the difficulties in estimating the available bandwidth along a path [Sanzgiri 2004]. These researchers transmitted probe messages to attempt to determine the amount of intra-flow interference. The results from these queries were used to compute what the authors call the “contention count” of each node. They define the contention count as the number of nodes on a multi-hop path that are within a node’s carrier sense distance. The researchers note

that computing contention count is challenging because the carrier sense distance can be longer than the distance over which two nodes can communicate. Nodes must be able to determine that they influence each other, even when they cannot directly communicate with each other.

2.2.2 Route Discovery

Route discovery protocols for ad hoc wireless networks can be reactive, proactive, or hybrid. Reactive protocols attempt to discover a route to the destination after data has arrived, while proactive protocols pre-compute routes to the destination. Hybrid protocols use a combination of the two approaches. Typically proactive protocols have lower latency for setting up connections, but require nodes to continuously exchange routing messages even in the absence of active flows.

2.2.2.1 QoS Routing Based on Bandwidth Estimation

BEQR (QoS Aware Routing Based on Bandwidth Estimation) [Chen 2005] nodes attempt to estimate their available bandwidth and broadcast this information as far as their two-hop neighbors. BEQR is based on ad hoc, on-demand distance vector (AODV) routing [Perkins 2003]. It uses a message containing the bandwidth requirements of a flow that is sent on multiple routes to the destination. If there is sufficient bandwidth, the request is forwarded to the next hop, otherwise the request is dropped. When the request reaches the destination, an acknowledgement is sent back along the reverse path. The same request messages are used periodically to perform adaptive feedback, potentially adjusting a route if a link is broken or its capacity drops too low to meet flow requirements.

2.2.2.2 OLSR and QoS Enhanced OLSR

Previously mentioned, OLSR is a proactive IP routing protocol designed for use on wireless ad hoc networks [Clauson 2003]. QoS Enhanced OLSR starts with the standard OLSR protocol and incorporates link bandwidth information to guide selection routes [Ge 2003]. In this approach, all of the routes are determined before use, just as in standard OLSR, but estimates of bandwidth available on links are used to favor higher bandwidth links. The bandwidth estimates are performed by monitoring the fraction of time the carrier sense mechanism of 802.11 detects that the channel is busy.

2.2.2.3 CEDAR

Non-QoS-aware routing is concerned primarily with finding a path from source to destination, but QoS-aware routing also takes into account additional factors such as the bandwidth or latency on links or the bandwidth required by a flow. In CEDAR (Core Extraction Distributed Ad Hoc Routing) [Sinha 1999], a subset of the nodes is dynamically elected to form a *core*, which is updated as network conditions change. Core nodes communicate with more than one other node, in contrast to edge nodes that communicate only with some core node. In this type of routing, the core nodes are selected to be the minimal set of nodes in which every node is either a core node or a one-hop neighbor of a core node. Routes are selected so as to pass through only core nodes except at the first and last hops. The core nodes exchange information on the current topology and link capacities. Since the number of core nodes is usually small compared to the total number of nodes, keeping track of this information is manageable.

When a node wishes to establish a connection to another node, it makes a request indicating the source, destination, and required bandwidth. Core nodes use broadcasts to discover a route to the final core node with each node adding its ID before rebroadcasting. When the message arrives at the final core node (which is either the destination, or a one-hop neighbor to the destination), the route is transmitted back to the initial core node. CEDAR will then attempt to find the shortest/widest path (least number of hops and highest excess bandwidth) that satisfies the bandwidth requirements for the flow. CEDAR is an example of a hybrid routing protocol in that some information required for routing is exchanged among core nodes before use, while some of the routing discovery is performed only after a flow arrives.

2.2.2.4 Multi-Channel Routing

Some researchers have also addressed the problem of multi-channel QoS routing [Tang 2005]. In this approach, the protocol must not only select a path but also choose a channel to use for each hop on each flow. Using multiple channels can reduce or eliminate self-interference across the hops of a flow.

These researchers present a linear-programming-based algorithm for computing an optimal path. Their approach employs a heuristic that is based on maximizing the "bottleneck capacity." They define the bottleneck as the link (on the selected channel) along a path where the spare capacity for a flow is minimal. Their heuristic attempts to find a path that maximizes this bottleneck.

However, their solution requires that significant information about the network topology, available bandwidth on each link, and currently active flows are known and that flows are splittable across multiple paths. One interesting result is that they show the shortest path does not always produce the best allocation. They demonstrate a 57% improvement in the probability that a path can be found for a newly arriving flow, compared to a shortest-path allocation scheme.

2.2.3 Resource Reservation

A QoS-sensitive application must have access to the resources it needs in order to run correctly. In wired networks, over-provisioning, or making available far more resources than necessary, is the most common approach. In wireless networks, resources are far scarcer since wireless networks typically have far smaller link capacities than wired networks. Different researchers have taken different approaches to assuring resources are available to QoS-sensitive applications in ad hoc wireless networks. Some have used explicit reservations, others use implicit reservations or priority, and still others rely on admission control and well-behaved applications (i.e., applications with specific predictable resource requirements).

2.2.3.1 Debt-Best Scheduling

One approach uses a combination of admission control and scheduling to make resources available to applications on networks in which links are characterized by a probability of dropping a packet [Hou 2009a, Hou 2009b]. These researchers provide an admission test based on long-term average throughput. According to their test, client's requirements are considered "fulfilled" if the long-term average (the limit as time goes to infinity) probability that the client transmits its target number of jobs in each scheduling interval is 1. The researchers further derive the necessary conditions for a set of tasks to be fulfilled under any scheduling policy that can be used as an admission control test.

In conjunction with the admission control test, the researchers provide two scheduling policies that they claim to be optimal (i.e., there are no other scheduling policies that can fulfill task sets that are not fulfilled by their policies). For the "largest time-based debt first policy," jobs in the queue accumulate debt as they wait and pay down debt each time a transmission is attempted until transmission is successful. For the "largest weighted-delivery first policy," the number of jobs that should have been accomplished by the current time is computed for each flow, and jobs are scheduled such that those for which this value is highest are transmitted first. The researchers use an ns-2 [Issariyakul 2008] simulation to validate their approach and show that both policies significantly improve timely throughput over the IEEE 802.11 DCF (Distributed Coordination Function) and EDCF (Enhanced DCF) standards. They also show that both policies eventually converge (i.e., find a steady state set of routes for flows) and satisfy the timeliness requirements of all flows that pass their admission test, but that the largest weighted-deliver first policy converges more quickly.

2.2.3.2 Ad Hoc On-Demand Routing

Ad hoc on-demand routing (AQOR) is a QoS-aware routing protocol that uses bandwidth reservation on the hops [Xue 2003]. In this approach, nodes advertise the amount of bandwidth they are using to their neighbors. Each node then uses the sum of its own and its neighbors' bandwidth consumption as an estimate for the total amount of bandwidth being consumed local by the wireless media. Admission control along a route is then based on this bandwidth estimation. Reservations in AQOR are auto releasing, with the reservation being dropped if no data is observed through a node for a specified period of time. By not requiring explicit messages to delete reservations, the network is better able to adapt when traffic is rerouted due to link failures. One of the difficulties is that the method used to estimate bandwidth utilization can overestimate consumption. This is because any traffic directly between two neighbor nodes will be counted twice. As a result, AQOR will tend to underutilize the network.

2.2.3.3 Alternative Metrics for Link Quality

Many researchers have noted that SNR (signal-to-noise ratio) is not a good metric for evaluating the quality of a link. One alternative that has been proposed [Han 2007] is USF (User Satisfaction Factor). USF is a function of the number of received bits and the delay sensitivity profile of an application. It is computed from an application-specific weight profile indicating the desirability of receiving data at a certain rate and a prediction of the number of bits that will be completed in the future. The application-specific weight profile can be defined based on the needs of the application and its response to reduced data transmission rates. At any point, the USF is computed by integrating the product of the weight profile and the bit-rate prediction from the current time to infinity. The authors propose four scheduling policies based on their USF:

- **Maximin:** always serves the job with the lowest USF first
- **Overall Performance:** attempts to maximize total USF across all jobs
- **Two-Step:** maximizes total USF while maintaining a minimum USF for each job
- **Proportional:** schedules according to a weighted ratio of the change in the USF a job would get if selected and its current USF

The researchers compare their approaches against weighted round-robin and proportional fairness policies, using a test data set with a mix of voice, video, and data streams. They show that Max-

imin results in a low USF difference among the different traffic types and that Overall Performance results in higher total USF. The authors are unable to show significant benefit of their approach over proportional fairness [Kelly 1998], other than reduced variance in the USF measurement over time. They also do not perform any sort of application adaptation in response to lower–than-required data rates observed by an application.

2.2.3.4 Implicit Reservations

Another approach to ensuring resources for flows is to use implicit reservation [Oh 2010]. In this method, an application sends a probe message to determine if the required resources are available prior to beginning a flow. At each node, the probe can be forwarded, pushed back, or rejected. Once a flow is established, it is given priority over any incoming flows or best-effort traffic. The authors propose a mechanism they call Neighborhood Proportional Drop (N-PROD) to enforce fair dropping of traffic across the flows. N-PROD is a distributed fair scheduling algorithm. As part of this algorithm, each node monitors the numbers of received packets and dropped packets to calculate its drop probability. Each node shares this information with its neighbors. Each node then computes a target drop probability for itself as the maximum of its local drop probability and the drop probabilities reported by its neighbors. When the node forwards a packet, it randomly drops packets to achieve its target drop probability. By doing this, congestion that would likely have occurred downstream can be avoided, resulting in more efficient use of resources.

Each flow is assumed to have a maximum tolerated drop probability. The assumed maximum is used to trigger rerouting when necessary. If the target drop probability at a node exceeds the maximum drop threshold for a flow, then "backpressure" pushes a flow back to the previous node for rerouting, in the same way as for originally establishing a flow.

2.2.3.5 Adaptive Modulation and Coding

Some researchers have investigated dynamic priority in order to maintain application QoS [Liu 2006]. In this approach, the priority of a flow depends not only on the application requirements, but also on the current quality that is being delivered to a flow. Their approach is based on the four QoS priority classes provided by the 802.16 standard:

- **Unsolicited Grant Service (UGS):** for constant bit-rate fixed throughput connections such as voice data
- **Real-time Polling Services (rtPS):** for connections needing throughput and latency guarantees such as video
- **Nonreal-Time Polling Service (nrtPS):** for connections needing throughput guarantees but not latency guarantees such as mission critical data transfers
- **Best Effort (BE):** for all other traffic not requiring guarantees

UGS traffic is always given preference, while the other three classes compete using a dynamic priority assignment algorithm. If there are no UGS packets to forward, a priority function is computed for each rtPS, nrtPS, and BE flow, and packets from the flow with the highest priority are selected for forwarding. The priority function includes a factor on whether a flow is falling behind in meeting its QoS requirements. When a flow begins to fall behind, its priority is increased and thus it is given preferential treatment.

2.2.3.6 Fair Queueing for Ad Hoc Wireless Networks

WFQ (Weighted Fair Queueing) is a popular approach for enforcing resource allocations in wired networks. It can be used to reserve portions of the bandwidth on a link to designated applications, when the link bandwidth is known and fixed. In wireless networks, it is difficult to directly apply WFQ since the medium is shared across multiple nodes that can not directly share information. One solution for this has been proposed by Luo [Luo 2004]. In this approach, slot times are computed for each flow with each slot having both start and finish times. The scheduler aims to choose the job with the soonest finish time from among jobs whose start time has passed. If no jobs are past their start time, then the job with the soonest start time is selected.

When applying this algorithm in a distributed environment, a backoff value is computed for each flow from information shared across nodes. When transmitting a packet for a flow, the router will wait for the backoff period to elapse before considering it eligible for scheduling based on its start and finish times. The backoff times are computed from the flow requirements and a flow contention graph. When new flows arrive or depart, a core router sends a multicast message back to all flow sources, updating the number of flows and their weights using a conflict-free minimum spanning tree to minimize the propagation time.

2.2.4 Route Maintenance

Since conditions in an ad hoc wireless network are subject to change, it is important to have methods to adapt to these changes. Adaptation can include decreasing the bit rate allocated to a flow or rerouting a flow to a more favorable route. To ensure that service meets QoS requirements, a scheme can rely on quick rerouting capabilities or on redundant resource reservation [Huang 2004].

2.2.4.1 Multi-Path Routing

One approach to maintaining high reliability routes is multi-SPEED routing protocol (MMSPEED) [Felemban 2005]. MMSPEED is designed primarily for sensor networks and assumes links with no retransmission. MMSPEED is based on geographic routing but can send a packet on multiple paths when required to meet reliability constraints that a flow has specified. Each time a packet is forwarded, MMSPEED estimates the probability that a packet will be successfully forwarded to the destination given that it uses each of several candidate next hops. It will then select enough next hops to meet the forwarding probability requirement for the flow. At each next hop, a new requirement is computed and the algorithm is repeated recursively.

In the event that the reliability requirements for a flow cannot be satisfied, MMSPEED will issue reliability backpressure messages to upstream nodes to downgrade the reliability expectations for those nodes. These backpressure nodes will trigger alternative routes to be found for future packets. Probability adjustments from backpressure messages time out after a specified time interval, based on the assumption that the condition that caused the backpressure message to be generated may have been resolved.

Another feature of MMSPEED is probabilistically guaranteed on-time delivery. A queue is maintained for each of several speed layers, with priority given to the highest speed layer. Each speed layer is characterized by a target geographic forwarding speed. The speed layer to be used by a packet is selected according to the geographic distance and the end-to-end latency requirement for

the packet. At each node and each speed layer, nodes calculate whether the target forwarding speed can be maintained. In the event that the target forwarding speed cannot be maintained, the node will probabilistically drop packets in order to maintain the target speed.

2.2.4.2 Redundant Path Reservations

Class-based approaches are primarily used for cell networks but are potentially applicable to ad hoc wireless networks [Huang 2004]. In these approaches, multimedia traffic is classified as either "real-time" or "non-real-time." Furthermore, handoff guarantees are classified guaranteed, priority, or best-effort. The goal of the approach is to maintain real-time connectivity through call handoffs (i.e., when the phone passes from one cell to the next). The researchers propose making tentative reservations on neighboring cells in addition to the primary reservation being actively used. For guaranteed connections, the tentatively reserved resources are available for use by the best-effort traffic until needed by the guaranteed connection. Tentative reservations are also made by priority connections, but the tentative reservations can be shared by two or more priority connections. As with those for the guaranteed traffic, the tentative reservations for priority traffic can be used by best-effort traffic until they are needed by a priority connection.

2.2.5 Route Selection

Route selection is the process by which a QoS mechanism chooses the route most likely to satisfy the QoS requirements of flows from among two or more alternative routes. Mechanisms for route selection must consider not only bandwidth and number of hops, but also factors such as latency and stability of the route. For example, a lower bandwidth route that is less likely to be disrupted might be preferred over a higher bandwidth route that more likely to be disrupted.

2.2.5.1 Adaptive Resource Allocation

In some wireless networks, ODFMA (Orthogonal Frequency Division Multiple Access) is used to enable multiple channels on wireless links. In such networks, different subcarriers are used to encode data to create the channels. Essentially, this results in providing multiple slower channels with the same total bandwidth as one fast channel. Some advantages of ODFMA over a single channel include shorter delay for lower bit-rate flows since there is less waiting, lower transmission power needed for low bit-rate flows, less contention since there are multiple channels available, and less susceptibility to multipath interference.

To effectively use an ODFMA system, there must be

- mechanisms to determine which subcarrier or subcarriers to assign to which flows
- a mechanism to determine the power level to assign to each subcarrier, in order to maximize throughput while minimizing loss and satisfying user QoS requirements

An exact solution can be formulated and solved for this problem using integer programming; however, this approach is not scalable to networks with more than a very small number of flows [Ergen 2003]. In addition to this exact solution, Ergen also compares several previously proposed approximate algorithms to a new iterative approximation algorithm. The iterative solution, Ergen shows, can find channel allocations with SNR (Signal to Noise Ratio) and power-per-bit similar to those obtained by the exact solution while significantly outperforming previous approximation algorithms.

2.2.5.2 Power Control

In networks where the power can be controlled per link, determining the amount of power to use on each link is a difficult problem. Keeping power use low can reduce interference on other links. One approach to this problem uses convex optimization methods to choose power levels for each link that maximize throughput while guaranteeing that QoS requirements are met [Julian 2002]. When transmitting data on a link, it is preferable to use just enough power to ensure that data is reliably received by the receiver. The researchers use the Signal-to-Interference Ratio (SIR) as the figure of merit. They formulate both a weighted fair power and minmax power optimization problem from SIR, while meeting throughput and delay requirements for each flow.

In addition to power optimization, the researchers implement an admission control and pricing scheme. The pricing scheme is based on how many "standard users" worth of resources the new user consumes in the system. The meaning of "standard user" is not explicitly defined by the researchers but is intended to be used simply as a reference point for "average" demand set by the system developer. The goal of this formulation is to price flows by the amount of lost future capacity they consume (e.g., a high-demand user might use 1.7 "standard users" of resources). Each flow added further constrains optimization. If the resulting problem is infeasible, the new flow cannot be accepted without affecting existing flows.

2.2.5.3 Network Utility Maximization

One recent approach to route selection for ad hoc wireless networks is based on NUM (Network Utility Maximization) theory [Bose 2008]. In this approach, the route selection problem is formulated as a convex optimization problem. Two types of flows are considered: (1) inelastic flows with a fixed bandwidth requirement and (2) elastic flows with a utility function specifying the user utility as a function of bandwidth. The researchers apply an optimization algorithm to find the optimal path and bandwidth to assign to each flow, though the algorithm assumes a utility function with specific mathematical form. In their model, multiple paths may be used to route traffic for a flow with portions of the flow bandwidth taking different paths. They also apply decomposition to break the optimization problem down into smaller sub-problems.

The researchers use simulation to demonstrate the value of their approach, but it is not clear that they are able to apply their optimization algorithm in a distributed manner. Furthermore, they are unable to achieve higher system utility in a fully loaded simulated tactical scenario compared to a baseline system without explicit optimization. They cite their assumption that they focus only on link bandwidth without consideration of the unreliable nature of wireless links as the cause for this result.

3 Adaptive Quality of Service (AQoS) Approach

Our AQoS approach combines previous methods to prioritize network data with immediate adaptive feedback to applications, without estimating or predicting future bandwidth. The key idea here is to provide a means by which applications can continue to carry out mission objectives by adapting to dynamically changing network conditions, specifically available bandwidth, without compromising the applications' critical runtime QoS.

AQoS embraces the dynamic, distributed, and unpredictable nature of ad hoc wireless networks. It assumes that failures and resource shortfalls are normal. As such, AQoS provides a framework allowing applications to adapt, rather than assuming rigid applications with the network attempting to adapt.

Adaptation has been tried in wired networks, where applications adapt to changing demand for resources as new application requests arrive and complete [Hoover 2001]. Requests are sent to a resource manager that evaluates the value of each application and sends messages to applications to adjust their level of service (e.g., frame rate or image resolution) in a way that optimizes the use of those resources. In small-to-medium-scale wired networks, the available resources do not change frequently and a centralized resource manager is a practical solution, but in ad hoc wireless networks this technique cannot be applied since available bandwidth and even network topology are constantly changing.

To address the issue of the changing environment in wireless networks, we use sensors in the network routers to detect and respond to congestion. We use backpressure techniques (similar in concept to research by Oh and Felemban [Oh 2010, Felemban 2005]) to notify upstream components of trouble spots in the network. Unlike previous approaches where the focus tends to be on rerouting around trouble spots, our focus is on application-level adaptation. Another way our approach differs from most previous approaches is that we do not attempt to directly estimate available bandwidth [Sinha 1999, Sanzgiri 2004].

AQoS demonstrates that the wireless routers used in mobile ad hoc wireless networks can sense network congestion and give notice to an application so that the application can adjust accordingly to avoid loss of service. AQoS detects congestion from its effects on queue lengths in the network routers.

Five techniques underlie the approach used for AQoS:

1. Each important flow of network traffic (i.e., flows of interest) is assigned a unique priority queue. Currently classification is performed on a wireless router similar to IntServ (discussed above in Section 2.1), but in future work the approach to classification will be performed at the source similar to that used in DiffServ (also discussed in Section 2.1).
2. Each queue for each classified flow is managed using traffic control facilities and the Hierarchical Token Bucket queuing discipline (htb qdisc) [Devera 2003] as a means for prioritizing use of available (but unknown) bandwidth based on the flow's priority. By using this mechanism, it is possible not only to prioritize flow traffic on the wireless link but also to observe the lengths of each queue.

3. Wireless routers in the network continuously observe the queue length for each flow and detect increased queue length as an early indicator of network congestion.
4. Upon detection of network congestion, a wireless router will provide feedback (in the form of a ***quench***[7]) message to the application(s) responsible for the flow of network traffic assigned to that queue. The feedback idea is similar to BEQR [Chen 2005] and CEDAR [Sinha 1999] in that the application is informed of network congestion and the application is expected to react. However, AQoS does not require bandwidth estimating.
5. On receipt of a quench message, an application can adapt to network congestion by selecting another QoS level that is intended to reduce network demand and contribute to reducing network congestion. This idea is similar to that seen in IntServ [Braden 1994] and in the Amaranth system [Hoover 2001], without the need for the applications to reserve bandwidth in advance or pre-register their profiles.

The ideas behind AQoS build upon a number of techniques and ideas identified in Section 2, combining them into an integrated approach. What is unique to AQoS is the ability for applications using the wireless network to be able to adapt to changing network conditions without the need to estimate or predict available bandwidth or use any bandwidth reservation scheme. To investigate and demonstrate AQoS, we created a model problem and experimental infrastructure for both laboratory and rudimentary field trials.

3.1 Model Problem

The problem chosen is illustrated in Figure 1.

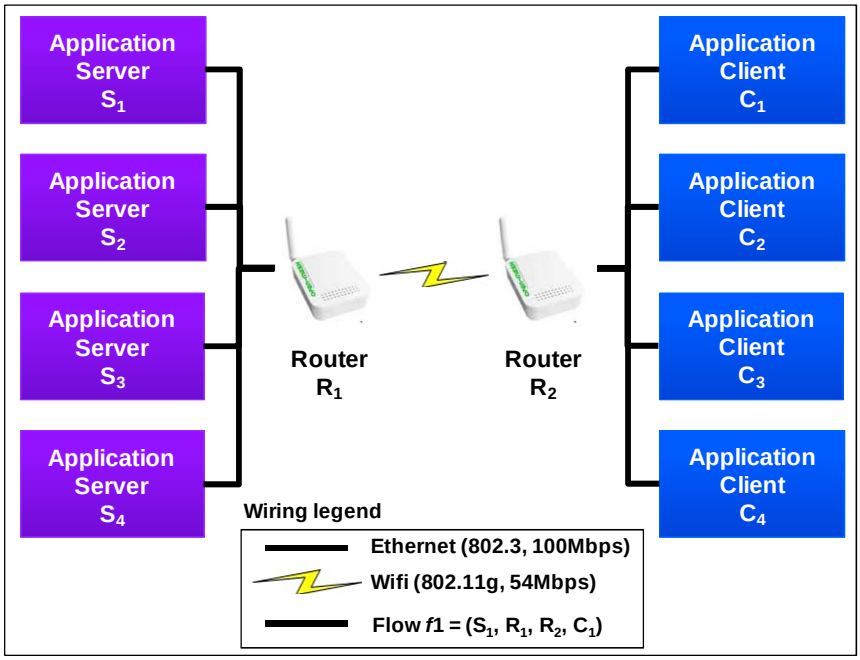

Figure 1: AQoS Model Problem Configuration

[7] For this report and in the context of AQoS, we use the terms *quench* and *quench message* as our form of feedback. This use should not be confused with RFC 792, which defines quench messages at the IP layer.

As indicated, there are a number of independent application servers (S_{1-4}) that provide information or data to peer clients (C_{1-4}). The servers and clients are connected to wireless routers by a wired connection. The routers themselves (R_{1-2}) are connected wirelessly using 802.11g and are participants in a mobile ad hoc wireless network using the OLSR routing protocol. Each flow between an application server and its client is differentiated from other flows between other servers and clients. For example, there is a flow defined by the path (S_1, R_1, R_2, C_1) and that flow is differentiated from the flow (S_2, R_1, R_2, C_2). All flows are unicast in nature. Contention must be recognized, and dealt with, by the routers as all flows transit the network routers, R_1 and R_2.

Using this configuration, each unique flow is thereby assigned a unique priority. Flow,

- (S_1, R_1, R_2, C_1) is assigned high priority (*f*1)
- (S_2, R_1, R_2, C_2) is medium high priority (*f*2)
- (S_3, R_1, R_2, C_3) is medium low priority (*f*3)
- (S_4, R_1, R_2, C_4) is low priority (*f*4)

Any other flows through the router are ignored and are relegated to a queue that is unmanaged and prioritized below the lowest priority flow of interest, *f*4.

The structure of this model problem has been used by a number of researchers in various investigations. For example, it was used in the development of the BLUE algorithm for active queue management [Feng 2001], algorithms to support use of priorities in video transmission [Chung 2002], and the simulation of active queue management in a QoS-enabled network [Koo 2005]. The configuration has also been used in the examination of multimedia services over the internet for developing a traffic-sensitive active queue management algorithm [Hwang 2010].

The model problem illustrated is extensible in a number of ways. For example, from a network configuration perspective, the number of application servers and clients may be varied, and additional wireless routers could be included. From a mission application perspective, the type of services conducted over the network can be varied as well as the specific metric used to measure QoS. There are also several variations that are possible from a software perspective, which are discussed in Section 5.

3.2 AQoS Assumptions

The initial approach for AQoS in the context of this model problem involves a number of assumptions. Foremost is the following: *In a mobile ad hoc wireless environment it is not possible to predict network characteristics (notably bandwidth) at any moment in time. However, it is possible to observe effects of the environment on the behavior of nodes, measure those effects, and then respond in some manner.*

Other assumptions include the following:

- Connectivity between the application source (S_n) and the router facing ingress to the wireless network (R_1) is wired. There are three reasons for this:
  a. Wireless communications from S_n to R_1 would cause interference between R_1 and R_2, which is beyond the scope of this initial investigation.

b. Quench messages sent from R_1 to S_n (being wired) would otherwise need to be prioritized over messages into the wireless network; therefore, quench messages would be in contention with application-level messages (from S_n to C_n).

c. The wired connection used here is lossless. In this manner, any loss measured will be attributed to the router facing egress to the wireless network.

- There is one application (S_n to C_n pair) per flow, and each flow has an associated unique priority that denotes the relative importance of that flow in relation to other flows. All priorities are statically assigned and do not change.
- Flows with higher priority use bandwidth before lower priority flows.
- For each flow, one application source (S_n) is responsible for the generation of network traffic occurring within that flow. (This does not mean network traffic cannot flow from C_n to S_n, but any such traffic would be minimal.)
- Quench messages are used by router R_1 to report network congestion. A quench message, similar in concept and purpose to that described in internet RFC 792,[8] is sent to the application responsible for the prioritized network flow (e.g., S_n) as a means to relieve network congestion.
- A router can detect network congestion and inform applications (via a quench), so that those applications can respond by altering their consumption behavior for bandwidth.
- Applications will faithfully alter their demand behavior for bandwidth in response to a quench message from a router.
- The protocol for application communication is UDP. TCP's built-in congestion management scheme would impede analysis of the AQoS approach at this time. Assessing AQoS against TCP flows is one aspect of future work discussed in Section 5.

Many of these assumptions are known to be limiting and fall short of any reasonable scale. However, the model problem as described above was constructed in such as way that many of the variables inherent in a mobile, ad hoc, wireless network could be controlled. In this way, we could focus on integrating past research and various techniques noted in the preceding discussion with AQoS to arbitrate application-specific, end-to-end QoS in an adaptive, cooperative manner. We will relax these assumptions in future work.

3.3 Applications in the Model Problem

Two applications were selected for the model problem. The first application is a variant of the Iperf[9] performance tool and the second is a video frame server (VFS). Each was selected for specific, experimental purposes described in the following discussion. Both applications exclusively use UDP as the communication protocol.

---

[8] See Section 3.2.2.3 Source Quench: RFC 792, readily available at http://www.freesoft.org/CIE/RFC/1122/43.htm as of December 2010.

[9] Iperf was developed by NLANR/DAST as a modern alternative for measuring maximum TCP and UDP bandwidth performance. Iperf allows the tuning of various parameters and UDP characteristics. Iperf reports bandwidth, delay jitter, datagram loss. For more information, visit http://sourceforge.net/projects/iperf/.

Iperf, an open source software component, was specifically designed to measure various aspects of IP-based network transmissions, including bandwidth (our focus). Most importantly, Iperf is able to maintain a constant and selectable bit rate for datagrams transmitted over any period of time. Datagrams transmitted by Iperf are also "small," where small is defined as a protocol data unit (PDU) size in bytes that is less than or equal to the maximum transmission unit (MTU). This characteristic is common (PDU <= MTU) in VoIP applications (e.g., small datagrams with specific timing requirements). Additionally, the behavior of Iperf could easily be changed to respond to a quench message from the wireless router in the model problem (discussed in Section 3.4.2.4) without embroiling us in the nuisances of VoIP at this stage of AQoS development.

The video frame server (VFS), developed in-house, interfaces via TCP to an off-the-shelf video camera and converts the TCP stream into UDP datagrams in MJPEG format [Wikimedia 2010] (and also adds timestamps and frame numbers to each UDP datagram—much like Iperf does to its packets). Each MJPEG frame produced by the VFS is different from the Iperf application in two aspects. For one, the size of each MJPEG frame is not constant from frame to frame due to JPEG compression and the video characteristics of each frame of video (compressed black frames will be much smaller in byte count than non-black, multicolored, frames). For another, the UDP datagrams transmitted by the VFS are "large," where large is defined as a PDU size in bytes that is greater than the MTU (i.e., PDU > MTU)—thereby resulting in network packet fragmentation. For AQoS to be effective, it must be able to contend with network flows that experience packet fragmentation. Like the modified version of Iperf, the VFS will also respond to a quench message from the wireless radio and adapt its behavior to network conditions (discussed in Section 3.4.2.4).

Both Iperf and VFS record the arrival of each datagram on the peer client for off-line analysis, the details for which are discussed in Section 4.

3.4 Routers in the Model Problem

The only discriminators for our selection of wireless routers were open source software and low-cost. The wireless routers selected for the model problem are the low-cost Open-Mesh OM1P Professional Mini Router[10] running the ROBIN open source mesh firmware (version r2690).[11] The ROBIN mesh firmware is a series of patches and new files augmenting the software source code from the open source project OpenWRT.[12] Using this software allowed for the unfettered inspection and, if necessary, modification of the software source code deployed on the radio.

As depicted in Figure 1, two OM1P routers were used in the model problem (R_1 and R_2). Router R_2, was used, basically,[13] as configured "out-of-the-box." However, to enable AQoS operation on Router R_1, three changes were necessary. First, we configured and installed the htb qdisc for the traffic control mechanism. Second, we installed an in-house developed QoS Management Agent

---

[10] For more information, visit https://www.open-mesh.com/store/products.php?product=Professional-Mini-Router.

[11] For more information, visit http://robin-mesh.wik.is/.

[12] For more information, visit http://openwrt.org/.

[13] That is, no changes were necessary to enable AQoS operation. All changes were specific localizations needed to identify the router as part of the local mesh network.

for Flow Control (QOSMA.FC) on Router R_1. The final change involved removing a single line of software source code that appears to have been introduced by the OpenWRT project[14] (see Appendix B).

The details of the htb qdisc and QOSMA.FC of AQoS are described in the following subsections.

3.4.1 Traffic Control and HTB Qdisc

Figure 2 depicts the hierarchical structure of the queues created on Router R_1 for the model problem. The htb qdisc used on Router R_1 has two key features:

1. It offers the ability to reserve bandwidth for different flows or queues and to share reserved and unused bandwidth between related queues. For htb qdisc, `rate` is the maximum bit rate a queue and all of its children are guaranteed (or reserved) and `ceil` is the maximum bit rate at which a flow represented by the queue can send if its parent has bandwidth to spare (unused bandwidth). If the `rate` and `ceil` are equal for a given child, a queue will not borrow unused bandwidth from its parent queue. However, if the `ceil` is greater than the `rate`, a queue is able to borrow unused bandwidth from its parent queue.
2. Child queues can be assigned a priority (`prio`) which simply means queues having unsent packets with a lower ordinal `prio` number (i.e., higher priority) are transmitted on the link first.

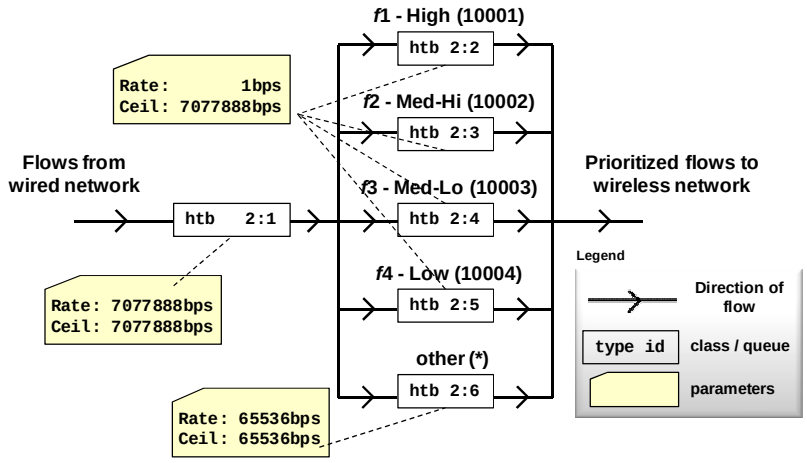

Figure 2: HTB Qdisc Configuration

What is unique about the AQoS approach is that **no bandwidth reservation** is actually specified (except for the unprioritized flows in queue 2:6 in Figure 2) when setting up the htb qdisc. **We use only the borrowing and priority features**. The parent htb qdisc (class 2:1) only specifies the known, theoretical, link capacity for 802.11g (54Mbit/s or 7,077,888 bytes/s). Child queues to this parent queue are created (class 2:2 through 2:5) to represent each of the four priority flows (*f*1 through *f*4, respectively) for the four prioritized applications in the model problem. The 5th queue (class 2:6) is for unprioritized flows.

[14] For more information, visit https://dev.openwrt.org/browser/branches/8.09/package/madwifi/patches/372-queue_vif.patch?rev=18054.

Each of the four child queues in Figure 2 is guaranteed (or reserved) only 1 byte/s (`rate` here essentially guarantees nothing) and is permitted to borrow up to the `rate` of the parent which is the link capacity for 802.11g. Furthermore, each child can borrow according to its assigned priority (class 2:2 for *f*1 first, and then class 2:3 for *f*2, and so on). Notice that unprioritized flows relegated to class 2:6 are not permitted to borrow (i.e., `rate = ceil`).

The commands used on Router R_1 to set up this structure and behavior appear in Appendix A.

3.4.2 QOSMA.FC

The QoS Management Agent for Flow Control (QOSMA.FC) is a process that executes on the Router R_1 as a user-level process (and therefore is not part of the kernel). This process uses the RTNETLINK library as a means to observe the state of the htb qdiscs discussed in Section 3.4.1. RTNETLINK is part of the iproute2[15] Linux kernel package.

Using the htb qdisc configuration described in Section 3.4.1, two states of the wireless network link become indirectly observable: undersubscribed and oversubscribed (or congested).

- Undersubscribed occurs when the total capacity required of the applications using the wireless network is less than that currently available on the network. In other words, there is sufficient bandwidth for all applications of interest to operate at the preferred, likely higher resource consuming, QoS. This is observed as all queues for the flows of interest are at or near zero queue length.
- Oversubscribed occurs when the wireless network's capacity is less than the total capacity required of the applications using the network. In other words, there is not sufficient bandwidth for all applications of interest to operate at the current QoS. This is observed when the queues for the flows of interest are filling up (i.e., queue length greater than or equal to some threshold). This is an early indicator of network congestion.

Actually, the behavior is more fine-grained than described above. Assume for the moment that all priority queues are receiving input at the same time and at the same bit rate. Lower priority queues will fill quicker than higher priority queues since high-priority flows being served first. That is, the queues grow in relative order from lower to higher priority queues.

3.4.2.1 QOSMA.FC Processing

Periodically, QOSMA.FC will poll the state of the installed priority queues. This period is configurable with the default being 250ms.[16]

- At each polling interval, the queue length[17] for each priority queue is recorded. Values are recorded for no more than the default of 10 intervals, after which older recorded values are discarded.

---

[15] For more information, visit http://www.linux-foundation.org/en/Net:Iproute2.

[16] Future versions of this agent will migrate from a polling-based pattern to an event-based pattern.

[17] Besides queue length, other RTNETLINK statistics are recorded including those classified under `TCA_STATS_BASIC`, `TCA_STATS_QUEUE`, and `TCA_STATS_RATE_EST`.

- Once recorded, the queue length for each priority queue of interest is tested. If the test (discussed in Section 3.4.2.2) evaluates to false, nothing happens. However, if the test evaluates to true, a quench message is dispatched to the application(s) responsible for the flow of network traffic assigned to that queue.
- After this processing is done, QOSMA.FC sleeps until the beginning of the next interval to start the processing all over again.

3.4.2.2 QOSMA.FC Tests

Currently QOSMA.FC supports any number of possible quench tests, with any individual test having any number of configurable parameters. To date, only four such tests were defined and are described here.

Table 1: Current List of QOSMA.FC Quench Tests

Quench Test Name	**If a quench message has not been sent in the last *t* ms, a quench message is dispatched ...**
Immediate	... if the observed queue length is greater than or equal to threshold, *n*
Over Time	... if the observed queue length is greater than or equal to threshold, *n*, for at least the previous *m* polling intervals
Immediate with Slope	... if the observed queue length is greater than or equal to threshold, *n*, and the queue is still growing
Over Time with Slope	... if the observed queue length is greater than or equal to threshold, *n*, for at least the previous *m* polling intervals and the queue is still growing

Each priority queue of interest can be assigned one quench test.

3.4.2.3 QOSMA.FC Quench Message Format

A quench message is dispatched as a UDP datagram to the application(s) responsible for the flow of network traffic assigned to that queue. For this model problem, only one application is assumed to be responsible for the prioritized network flow (also discussed in future work in Section 5). The content of that quench message is described in Table 2.

Table 2: Description of Quench Message Fields

Field Name	**Description**
`cto`	Expiration time for the quench message, in milliseconds (ms). This is the number of ms when the quench message expires after receipt, and the application is free to restore QoS to a previous setting.
`quid`	Quench ID. This is an integer that is initially zero and is incremented on each subsequent, new, quench message (that is when the test executed in Table 1 evaluates to true). A quench ID may be repeated if it is sent from the QOSMA.FC more than once over the UDP connection for purposes of reliability.
`guid`	Globally Unique ID. This is the ID of the Router that sent the quench message. The only requirement is that the value be globally unique.

3.4.2.4 QOSMA.FC Application Response to a Quench

When the application responsible for the flow of prioritized network traffic receives a quench message, the application can either

- ignore the quench[18] and continue processing as before
- react to the quench by changing its behavior

By ignoring the quench, the application is choosing to continue without changing its behavior in the face of changing network conditions. As such, the application may soon or will continue to experience packet loss as a result of congestion. Taking this course of action, the treatment Router R_1 provides the application is nothing more than that which can be provided through the use of the priority queues on Router R_1. This will likely have no adaptive effect of reducing the demand for network resources.

However, by choosing to react to the quench message, the application is adapting to changing network conditions and reducing its demand for network resources. The means by which the application adapts is solely at its discretion and takes into consideration its mission and requirements. Although such adaptation may not eliminate packet loss, the adaptation will likely reduce loss in exchange for a reduced data rate.

QOSMA.FC includes a quench message timeout (see `cto` in Table 2) in anticipation that network conditions will improve. The application uses this timeout to return to its behavior prior to the most recent quench. Should the application revert to its prior behavior and network conditions not improve, the application will receive another quench message from the QOSMA.FC, repeating the adaptive QoS process.

The two applications included in the model problem, Iperf and VFS, respond to a quench message by performing the following behavior:

- Iperf reacts by reducing its bit rate by 500,000 bits/s. Upon a timeout, Iperf increases its bit rate by 100,000 bits/s. If another timeout period expires without receipt of an additional quench message, Iperf again increases its bit rate by 100,000 bits/s until it reaches its preferred QoS bit rate.
- VFS reacts by reducing its bit rate by selecting the next lower QoS level as defined in Table 8 on page 34. Upon a timeout, VFS increases its bit rate by selecting the next higher QoS level. If another timeout period expires without receipt of an additional quench message, VFS again increases its bit rate by one QoS level until it reaches its preferred QoS level.

In both cases, should another quench message arrive before the timeout expires, the old timeout is discarded and the new timeout value is used.

The behavior described here is only one of any number of possible responses to a quench message. It is intended to be simple, uncomplicated, and sufficient for evaluating the AQoS network management policy against other policies (described in Section 4.1).

[18] As per the assumptions stated earlier, this course of action is not taken in the model applications.

4 Experiments and Results

The goal of these experiments was to quantitatively demonstrate the effects of AQoS compared to other traffic management policies in response to dynamic, real-time variations in bandwidth caused by interference in the mobile wireless environment. To quantify such effects, applications were subjected to bandwidth undersubscriptions and oversubscription scenarios to measure application data rate and packet loss.

For these experiments, we defined *progressing with tolerable packet loss* as the application being able to serve requirements, albeit at a lower QoS level, in the face of packet loss. Since the applications used in these experiments were only model applications (for reasons discussed in Section 3.3), this notion was artificial in that the only mission requirement these applications had was to record the receipt of packets. Our objective in using AQoS was to see that the applications, Iperf and VFS, operate at or as close to their preferred QoS level as long as possible, while at the same time incurring the lowest possible packet loss given network conditions.

4.1 Methodology

Three different traffic management policies for the routers were defined to investigate the effects of AQoS:

1. *No Management*—This policy (Policy 1) is simply the default "out-of-the-box" configuration for both routers in Figure 4 and Figure 5; it provides no special treatment of flows.
2. *Prioritized Flows*—This policy (Policy 2) changes the configuration of the Router R_1 to use the htb qdisc as discussed in Section 3.4.1 to direct flows to the associated priority queues. There is no quenching in this policy.
3. *AQoS with Immediate Test*—This policy (Policy 3) adds to the prioritized flows in Policy 2 by using the QOSMA.FC, as discussed in Section 3.4.2, on Router R_1's configuration to monitor and if necessary quenching the application responsible for the prioritized network flow.

Each of the applications (Iperf and VFS) was tested in both the laboratory and the field under each of the defined policies. Key in both experimental settings was to set conditions by which packet loss would occur through oversubscription of the wireless network.

In ideal conditions, the wireless radios in the two routers were expected to operate at 54Mbit/s. Since 802.11g uses the same wireless channel to both transmit and receive, the expected capacity of the wireless network in these conditions for our applications would be on the order of 22-25Mbit/s[19] (accounting for 802.11g protocol overhead) [Proxim 2003]. In such ideal conditions, both applications with their respective flows should be able to operate at their preferred QoS level (Table 5 and Table 7), as the cumulative bandwidth demands are on the order of 5Mbit/s for Iperf

[19] 27Mbit/s is the throughput for either *transmit* or *receive*, taking into account all symbols in the wireless stream. The throughput realized by an application after taking into account all medium, IP and UDP protocols is thereby reduced.

and 18Mbit/s for VFS. The cumulative demand for each of the two applications is less than the expected capacity of 22-25Mbit/s, and therefore the wireless network is undersubscribed.

To achieve oversubscription in the wireless network, our approach was to get the two wireless radios to operate at a lower data rate. Under 802.11g, radios will dynamically operate at lower or higher data rates to compensate for weaker or stronger radio signals. Such dynamic perturbations are common in 802.11g wireless networks, and the stepped data rates for 802.11g are shown in Table 3.

Table 3: 802.11g Data Rates

802.11g Data Rates (Mbit/s) for 2.4GHz Band											
1	2	5.5	6	9	11	12	18	24	36	48	54
weaker signal <----------> stronger signal											

In the laboratory, oversubscription was accomplished by manually programming the radios to operate at lower data rates to simulate a weaker signal and, therefore, control the specific data rate used by the radios for various tests. This was necessary because it was not possible to separate the radios by a distance sufficient to get the lowest data rate. By programmatically using these diminishing 802.11g data rates, it was possible to oversubscribe the wireless network to a desired, and controlled, severity of oversubscription (i.e., the lower the data rate the more severe the oversubscription).

In the field, oversubscription was accomplished by incrementally increasing the distance between the two wireless radios until the signal was so weak that the connection between them was lost (i.e., where no data packets could be received). As the radio signal weakened, the effect was a lower data rate that, as in the laboratory, resulted in oversubscription. Unlike the laboratory, however, the specific data rate automatically selected by the radios in this environment was not controlled.

We observed that the wireless network noticeably began to experience oversubscription at 0.10 miles from the base station (specifically marker B in Figure 3), became profoundly oversubscribed at 0.15 miles (marker C), and generally lost connectivity at 0.20 miles (marker D). All except one of the field experiments was conducted between the base station and the marker labeled as *east turn*. In the Continuous VFS Experiment (conducted as a circular route from the base station to the east turn, west turn, and back to base), oversubscription would also become profound soon after 0.05 miles (marker E), due to blocking by the Earth when the line-of-sight between the base station and the vehicle used in the field was lost.

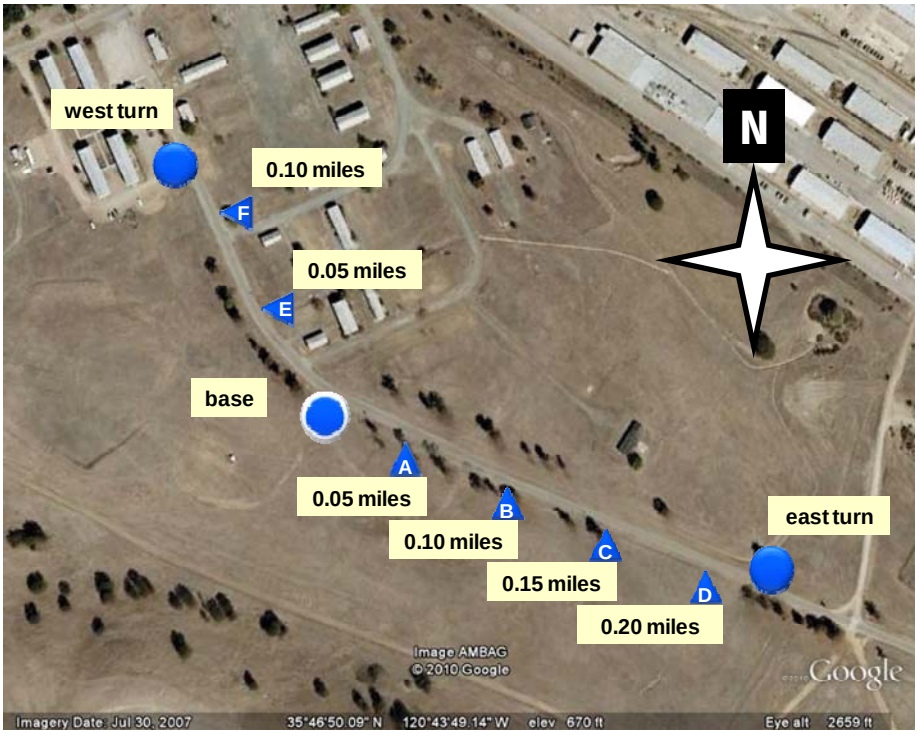

Figure 3: Field Test Driving Course

4.2 Design and Expectations

The experiments were designed for these two basic scenarios, undersubscription and oversubscription. For undersubscription scenarios, it was expected that under all policies (defined in Section 4.1) all applications would be able to progress with tolerable packet loss. The hypothesis for undersubscription scenarios was that AQoS would perform as well as the other traffic management policies defined for these experiments and *no worse*.

However, for the oversubscription scenarios, expectations differed depending on the traffic management policy in effect.

- *Policy 1, No Management*—All application flows would experience loss, equally, without regard to the application flow's priority. That is, no preferential treatment would be given to one flow over another, and all applications would begin to experience packet loss and fail to progress at the same time. This would be typical of unmanaged (wireless) networks.
- *Policy 2, Prioritized Flows*—Application flows would experience loss in reverse order of the application flow's priority. That is, the Router R_1 would give preferential treatment to higher priority application flows over lower priority flows, and higher priority applications would be able to progress with tolerable packet loss longer than lower priority flows as network bandwidth diminished. This would be typical of QoS mechanisms using only priority queues.
- *Policy 3, AQoS with Immediate Test*—Like Policy 2, application flows would experience loss in reverse order of the application flow's priority. However, flows having lower priority would receive quench messages first, so that those applications could change their behavior

and demand less bandwidth. This would allow those lower priority application flows to progress in the face of diminished network bandwidth. The hypothesis here was twofold:

a. Lower priority flows are able to progress longer than that seen in Policy 1 or 2 but will eventually fail once the wireless network can no longer support any demand for bandwidth required by these lower application flows.

b. Higher priority flows are protected or isolated from diminished network bandwidth effects longer than other lower priority flows until the wireless network can no longer support the preferred data rate of the highest priority flow.

The experiments conducted for this report are summarized in Table 4. Each experiment was conducted (and data collected) for each policy defined above, resulting in 18 individually conducted experiments. The results are reported and discussed in Section 4.4.

Table 4: Summary of Experiments Conducted

Experiment	Location	Type	Application	Control Variable	Section
1	Laboratory	Stationary	Iperf	802.11g Mode	4.4.2
2	Field	Discrete	Iperf	Distance	4.4.3
3	Laboratory	Stationary	VFS	802.11g Mode	4.4.4
4	Field	Discrete	VFS	Distance	4.4.5
5	Field	Stationary	Iperf	Application Bandwidth	4.4.6
6	Field	Continuous	VFS	Speed and Distance	4.4.7

The details and nuances for the control variable's settings are discussed along with the reported results in the following section. The *Type* column in Table 4 is defined as follows:

- *Stationary*—Routers in the setup (and the computers wired to those routers) were not moved during the duration of the experiment and remained a fixed distance apart.
- *Discrete*—Only one of the routers (R_1) in the setup was mobile and its distance from the other router in the setup was discretely controlled and moved at specific times in the experiment.
- *Continuous*—Again, only one of the routers in the setup was mobile. However the distance from the other router in the experiment was continuously changing at a constant rate for the duration of the experiment.

For the Discrete and Continuous experiments conducted in the field, the mobile router and associated computers (see Setup below) were placed in a vehicle.

4.3 Setup

The model problem illustrated in Figure 1 was instantiated for the laboratory and field settings. The laboratory setting (shown in Figure 4) was simply two computers, configured to run either the Iperf application or the VFS application and wired to an Open-Mesh OM1P Professional Mini Router. When performing the Iperf tests, the computers were configured to run a Linux/Ubuntu distribution. Otherwise, when running the VFS tests, the computers were running Microsoft Windows XP or Windows 7.

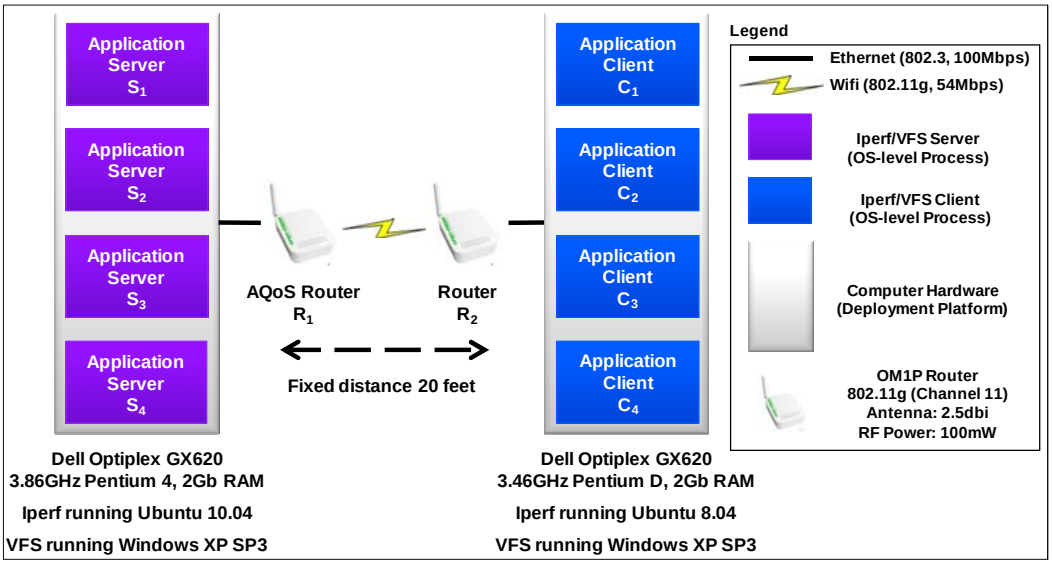

Figure 4: Laboratory Setup

In either case (Iperf or VFS experiments), the computer configured to be the server machine was connected to the modified AQoS Router R_1 (left in Figure 4), while the other client machine was connected to the unmodified Router R_2 (right in Figure 4). These two computers with their respective, attached, wireless routers were placed, arbitrarily, 20 feet apart in the laboratory, an indoor facility at the SEI. This short distance was selected to achieve a wireless link with virtually no loss.

The second instantiation of the model problem for the field setting was nearly identical to the laboratory setup with a few major differences to support mobility (see Figure 5). The computer configured to be the server machine (and connected to the AQoS Router R_1) was placed in a vehicle. The computer configured to be the client machine remained at a stationary base station (see Figure 3).

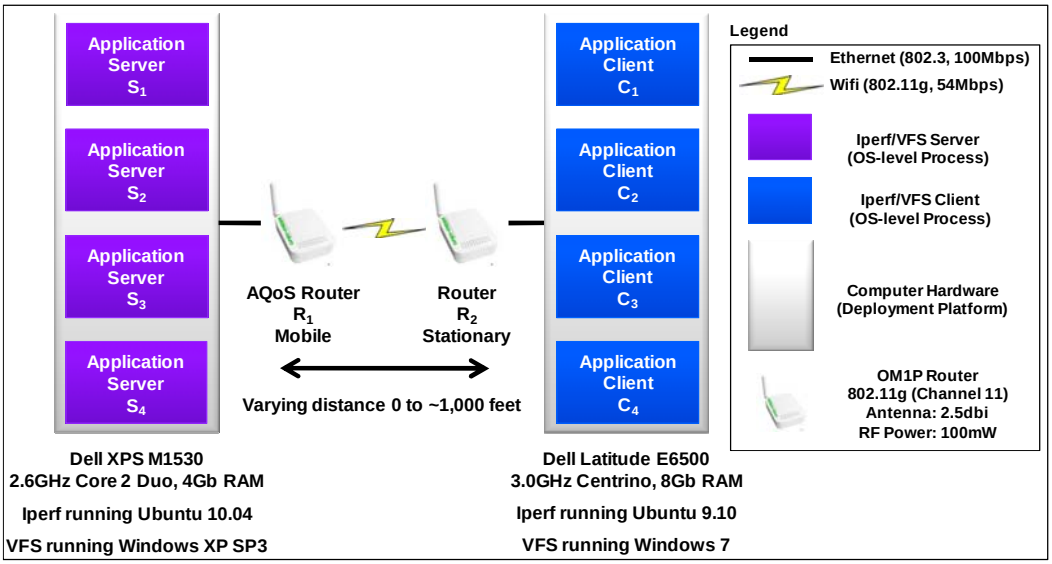

Figure 5: Camp Roberts (Field) Setup

The field setup was conducted at the NPS Center for Network Innovation and Experimentation[20] (CENETIX) testbed called the Tactical Network Topology (TNT)—which occurs at Camp Roberts, California on a quarterly basis. The course used to conduct the Iperf and VFS tests using the configuration in Figure 5 was a two-tenths of a mile road at Camp Roberts (35°46'49.56"N, 120°43'53.57"W) with little vegetation, other 2.4GHz sources, and terrain obstacles[21] (see Figure 3).

4.3.1 Application Settings

Between the two applications used in the model problem and the number of tests available for use with QOSMA.FC, there are many permutations of settings that could have been used for the model problem. The parameters used for Iperf (see Table 5), VFS (see Table 7), and QOSMA.FC (see Table 10) were not scientifically determined. Rather, they were set according to our best judgment given the intent of comparing the AQoS Policy 3 to the other two traffic management policies.

4.3.1.1 Iperf

UDP port numbers 10001 through 10004 were used to indicate data stream flow priority. Destination port 10001 indicated the highest priority data (flow *f*1); port 10004, the lowest (flow *f*4). These port numbers were used by Router R_1 to assign flows to the associated priority queues (see Figure 2).

Data was sent from the Iperf server (left in Figure 6) to the Iperf client (right in Figure 6) at a predesignated (or initial) data rate, with the highest priority flow (*f*1) having the lowest data rate of 0.5Mbit/s and the lowest priority flow (*f*4) having the highest rate of 2Mbit/s. The cumulative bandwidth needed by all the Iperf clients was a constant 5Mbit/s. Table 5 details all flows.

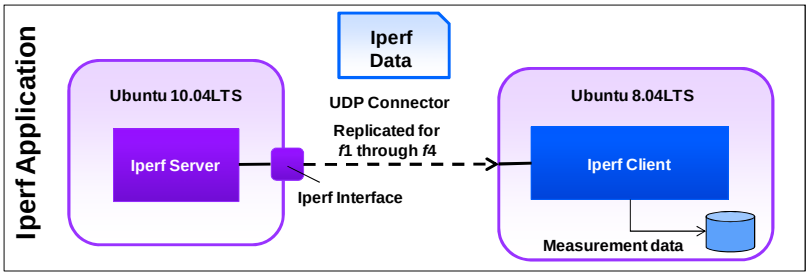

Figure 6: Iperf Application Configuration

Table 5: Iperf Data Stream Parameters (Default)

Flow	Priority	Data Rate [bit/s]	UDP Port
*f*1	1 (highest)	512,000	10001
*f*2	2	1,024,000	10002
*f*3	3	1,536,000	10003
*f*4	4 (lowest)	2,048,000	10004

[20] See http://cenetix.nps.edu/cenetix/.

[21] Although terrain was introduced in the "Continuous VFS Experiment" discussed later.

As discussed in Section 3.4.2.4, Iperf's response to a quench message would be to reduce its data rate by 500,000 bits/s. Then, after the quench timeout (`cto` in Table 2), Iperf would increase its data rate by 100,000 bits/s back to the pre-designated data rate. For all flows, regardless of the pre-designated data rate, the lowest data rate is 10,000 bits/s. (The data rate will never be zero.)

As data is sent from the Iperf server to the Iperf client, the client will log data of the transfer to a file on disk. The fields in that log file are described in Table 6.

Table 6: Iperf Measurement Log Data

Field Name	Description
`Packet Number`	Packet number of the individual Iperf datagram. This integer is initially zero and monotonically increases
`Sent Time`	Time stamp inserted by the Iperf server indicating the time when the individual Iperf datagram was sent (in ms)
`QoS Level`	QoS level under which the Iperf server was operating, in bits/s (never less than 10,000)
`Received Time`	Time stamp recorded by the Iperf client indicating the time when the individual Iperf datagram was received (in ms)

4.3.1.2 VFS

For VFS, the same UDP port numbers 10001 through 10004 were used to indicate the priority of data streams flows, to differentiate the flows, and to assign the flows to the associated priority queues.

Likewise, VFS data was sent from the VFS server (left in Figure 7) to the VFS client (right in Figure 7) at a pre-designated (or initial) data rate. However, in this case, all flows had the same data rate of 4.5Mbit/s. (All flows are detailed in Table 7.) Therefore, the cumulative bandwidth needed by all the VFS clients was a constant 18Mbit/s.

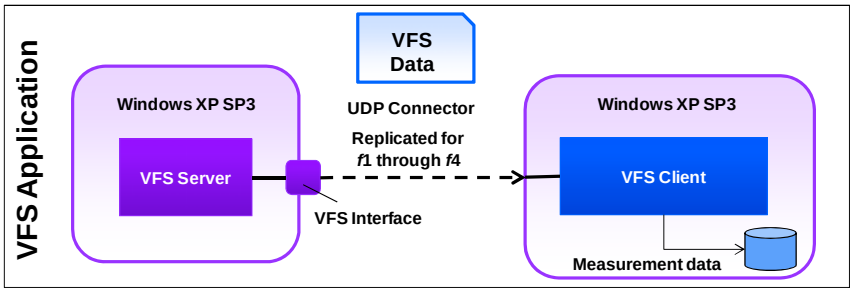

Figure 7: Video Frame Server (VFS) Application Configuration

Table 7: VFS Data Stream Parameters (Default)

Flow	Priority	Data Rate [MBit/s]	Frame Rate (fps)	UDP Port
*f*1	1 (highest)	4.5	25	10001
*f*2	2	4.5	25	10002
*f*3	3	4.5	25	10003
*f*4	4 (lowest)	4.5	25	10004

Similar to Iperf, VFS would respond to a quench message by reducing its data rate; however, VFS would select a lower data rate by changing one or both parameters of the MJPEG data stream

(JPEG quality and resolution, shown in Table 8). Like Iperf, VFS would restore itself to a higher QoS level after the quench timeout.

Table 8: Five QoS Levels for VFS

QoS Levels	Data Rate [MBit/s][22]	fps	JPEG Compression Quality	JPEG Resolution (in pixels)
4 (default)	4.5	25	85%	Full (480h x 250w)
3	1.6	25	85%	1/2 Full
2	0.8	25	50%	1/2 Full
1	0.4	25	70%	1/4 Full
0	0.2	25	50%	1/8 Full

The VFS client also logs data of the MJPEG frames received to a file on disk. The fields in that log file are described in Table 9.

Table 9: VFS Measurement Log Data

Field Name	Description
`Frame Number`	Frame number of the individual MJPEG received. This integer is initially zero and monotonically increases.
`Sent Time`	Time stamp inserted by the VFS server indicating the time when the individual MJPEG frame was sent (in ms)
`Frame Size`	Size of the individual MJPEG frame (in bytes)
`QoS Level`	QoS level under which the VFS was operating, values 0 - 4, see Table 8
`Received Time`	Time stamp recorded by the VFS client indicating the time when the individual MJPEG frame was received (in ms)

4.3.2 QOSMA.FC Settings

In Section 3.4.2.2, QOSMA.FC could execute one of four tests to determine if a flow had to be quenched. For all experiments, and all application tests (for all flows), the test used was *Immediate*. The parameters for that specific test are shown in Table 10.

Table 10: Parameters Used in the Immediate Test for the Experiments

Parameter	Value	Description
`Threshold`	`21`[23]	Minimum value of a flow's queue length which if exceeded triggers a quench
`Timeout`	`2000`	Minimum time interval between two successive quench messages (in ms)
`cto`	`2000`	Defined in Table 2 on page 24

As a reminder, QOSMA.FC was configured to observe the state of the priority queues at a rate of four times per second or once every 250 milliseconds (see Section 3.4.2.1).

[22] The data rate reported here is the average data rate of the MJPEG data stream used in the experiment as measured by Wireshark (http://www.wireshark.org/).

[23] The size of the queues used in Linux's queuing disciplines is 127 packets (or skb: socket buffers). The value 21 was selected as it is between 15-20% of the queue size, simply a judgment on our part.

4.4 Results

4.4.1 Understanding the Graphs

For each of the figures presented in the following sections (one figure for each of the six experiments), three graphs are shown (one graph each for the three traffic management policies). The control variable for each experiment varies along the X-axis for each graph and is discussed in the details below for each of the experiments.

The Y-axis represents data loss and the application data rate for each of the four prioritized flows. Both loss and data rate are determined by collecting data about sent and received packets in the data streams for a period of time.

1. Loss is defined as $1 - \frac{P_r}{P_s}$ where
   - P_r is packets received, and
   - P_s is packets sent.
2. Average Application Data Rate is plotted as a fraction of the preferred data rate, that is $\frac{1}{PDR} \cdot \frac{1}{T} \cdot \int_0^T ADR(t)\, dt$, where:
   - T is overall observation time (e.g. 60s),
   - PDR is the flows preferred data rate (the default from either Table 5 for Iperf or Table 7 for VFS), and
   - $ADR(t)$ is the application data rate at time t (i.e., the data rate selected by the application in response to quench messages as described in Section 3.4.2.4).

For Policies 1 and 2, the percentage of Application Data Rate will remain at 100% as the applications under this policy are not quenched and therefore will always transmit at their preferred data rate. However, applications subjected to Policy 3 will transmit at different data rates since they react to quenching under this policy, and therefore the Percentage Application Data Rate will be either at 100% or some percentage of their preferred data rate.

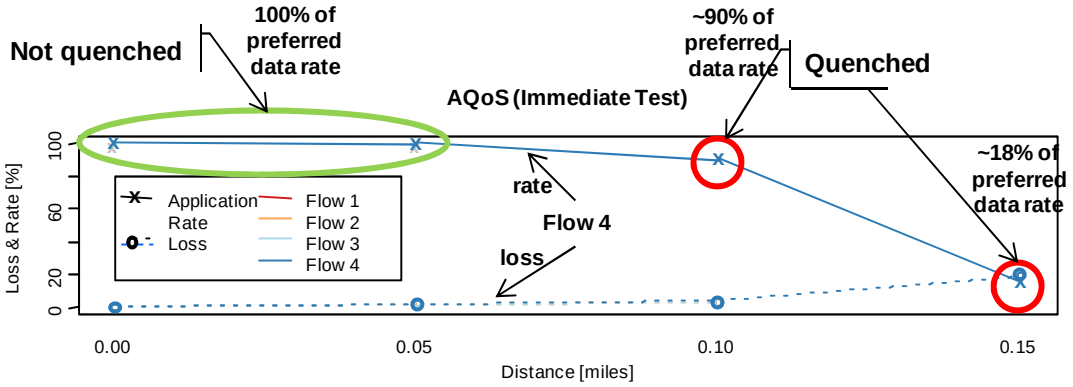

Figure 8: Instructional Graph

To illustrate, Figure 8 shows an example of one of the experiments conducted for Policy 3 (AQoS with Immediate Test). In this example, the X-axis is the distance between the two wireless radios. The Y-axis is percentage loss (the dashed line with circles at each data point) and percentage application data rate (the solid line with crosses as each data point).

The experiment was conducted by placing the radios at different distances apart (first at 0.0 miles apart, then 0.05 miles apart and so on) and then executing the application (Iperf in this case). At each placement, data was recorded for each flow. This data was used to create the graph. To make it easier to explain the graph in Figure 8, only one flow is depicted, Flow 4 (the blue line). In the following results, all flows are depicted on the same graph for each policy.

Consider the two lines depicted for Flow 4 in Figure 8:

1. *Percentage Loss*—As the two radios get further apart, the percentage loss increases from nearly 0% loss (at distances 0.0 and 0.05 miles) to nearly 20% loss at 0.15 miles.
2. *Percentage Application Data Rate*—With the radios placed 0.0 and 0.05 miles apart, the application was able to maintain its preferred data rate (which for Flow 4 is 2Mbit/s), plotted at 100%. Since traffic management Policy 3 (AQoS) was in effect and the flow is at 100% of its application data rate, it can be deduced that there was sufficient network capacity to support this flow (i.e., undersubscribed) and that no quenching was necessary for this flow. In the subsequent remaining radio placements (0.10 and 0.15 miles apart), this was not the case. At both placements, Flow 4 was quenched, meaning that there was not sufficient capacity to operate Flow 4 at its preferred data rate (i.e., oversubscribed) and that quenching was necessary. At 0.10, quenching Flow 4 resulted in a realized data rate which was approximately 90% of its preferred data rate or ~1.8Mbit/s. This is further reduced to approximately 18% of its preferred data rate at 0.15 miles (or 0.04Mbit/s).

Taking these two plots together for Flow 4, the inference from this is that from 0.0 to 0.5 miles, this lower priority application (Flow 4 is the lowest priority flow) is able to make progress with no loss at its preferred data rate, thereby achieving the applications preferred QoS. However, at greater distances, and in order to reduce loss yet still make progress, this lower priority application must reduce its data rate from its preferred rate to a lower rate to incur less loss—which it does under Policy 3—to the point of transmitting at a much lower rate when at 0.15 miles (than at either of the shorter distances) to keep loss to a minimum (here approximately 20%).

4.4.2 Experiment 1—Iperf in the Laboratory

Undersubscription and oversubscription for this experiment was achieved through programmatically setting the control variable to a fixed 802.11g data rate. The four flows operated under each of the policies for one minute duration. The Iperf client recorded and logged the data from the session for 60 seconds for each mode selection. Those data are depicted in Figure 9.

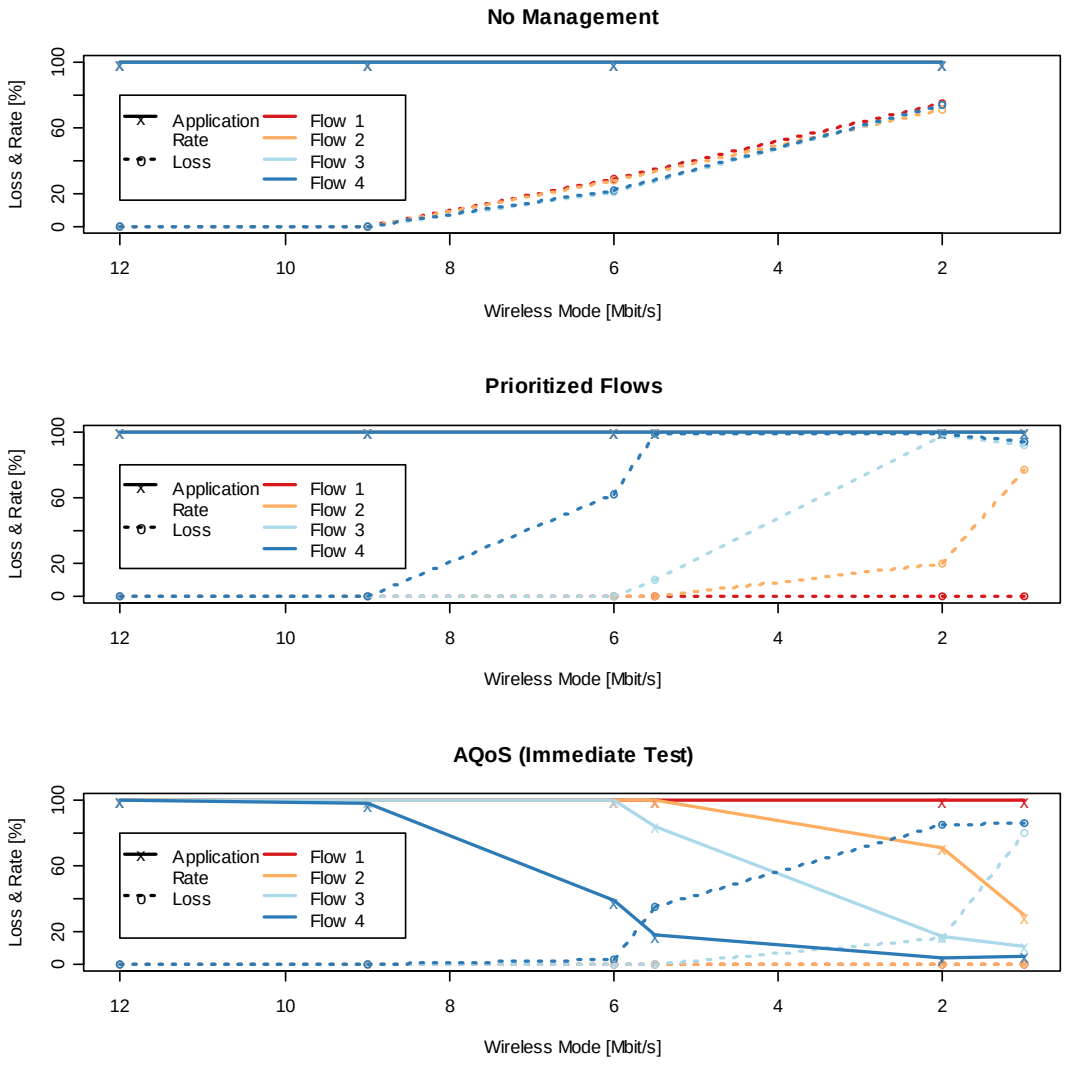

Figure 9: Iperf Stationary Tests in the Laboratory

Note: The lines that connect the data points in this figure only highlight the related points for each of the flows represented in the graph and cannot be used to interpolate data rates between those set for this experiment.

The first and second data points on the X-axis, 12Mbit/s and 9Mbit/s, show that all Iperf applications are able to progress with no loss. This is consistent with the fact that all the applications and their associated flows cumulatively consume 5Mbit/s, which is under the capacity of the wireless network.

All flows are impacted in the same way when the available bandwidth is reduced under Policy 1 (i.e., No Management, top graph in Figure 9). Policy 2 (i.e., Prioritized Flows, middle graph in Figure 9), leads to loss affecting flows according to their priority. While operating under Policy 3 (i.e., AQoS with Immediate Test, bottom graph in Figure 9), quenching shifts the impact to the right. Policies 2 and 3 are able to protect the highest priority flow because the lowest available bandwidth on the wireless link (approx. 0.5Mbit/s in wireless mode 1Mbit/s) is enough to transmit

the required 0.5Mbit/s high-priority application data rate. Additionally, Policy 3 is able to reduce data loss for the flow with the second highest priority by reducing the application data rate for this flow. The remaining two flows are quenched to a very low data rate because of insufficient bandwidth. The fourth flow (lowest priority flow) operates at just under 100% of its preferred data rate, meaning that during the course of the execution of this experiment it did react to a quench from the AQoS Router R_1.

4.4.3 Experiment 2—Iperf in the Field

Undersubscription and oversubscription for this experiment were achieved by increasing the distance between the two wireless routers at discrete intervals. In this experiment, the vehicle with the server equipment (the left side of Figure 4 on page 31) was parked at the base station (0.0 miles in Figure 3 on page 29), marker A (0.05 miles), marker B (0.10 miles) and marker C (0.15 miles). The four Iperf application flows were operated for 10 minutes at each marker. These data were recorded and are shown in Figure 10.

There is a subtle difference between the Iperf application experiment conducted in the laboratory and the field experiment discussed here. In the laboratory, the distance between the wireless radios was fixed and very short, the assumption being that no real loss on the wireless link would occur between the two radios in the laboratory due to their proximity. In fact, this assumption also turned out to be true based on the observations from the field experiment.

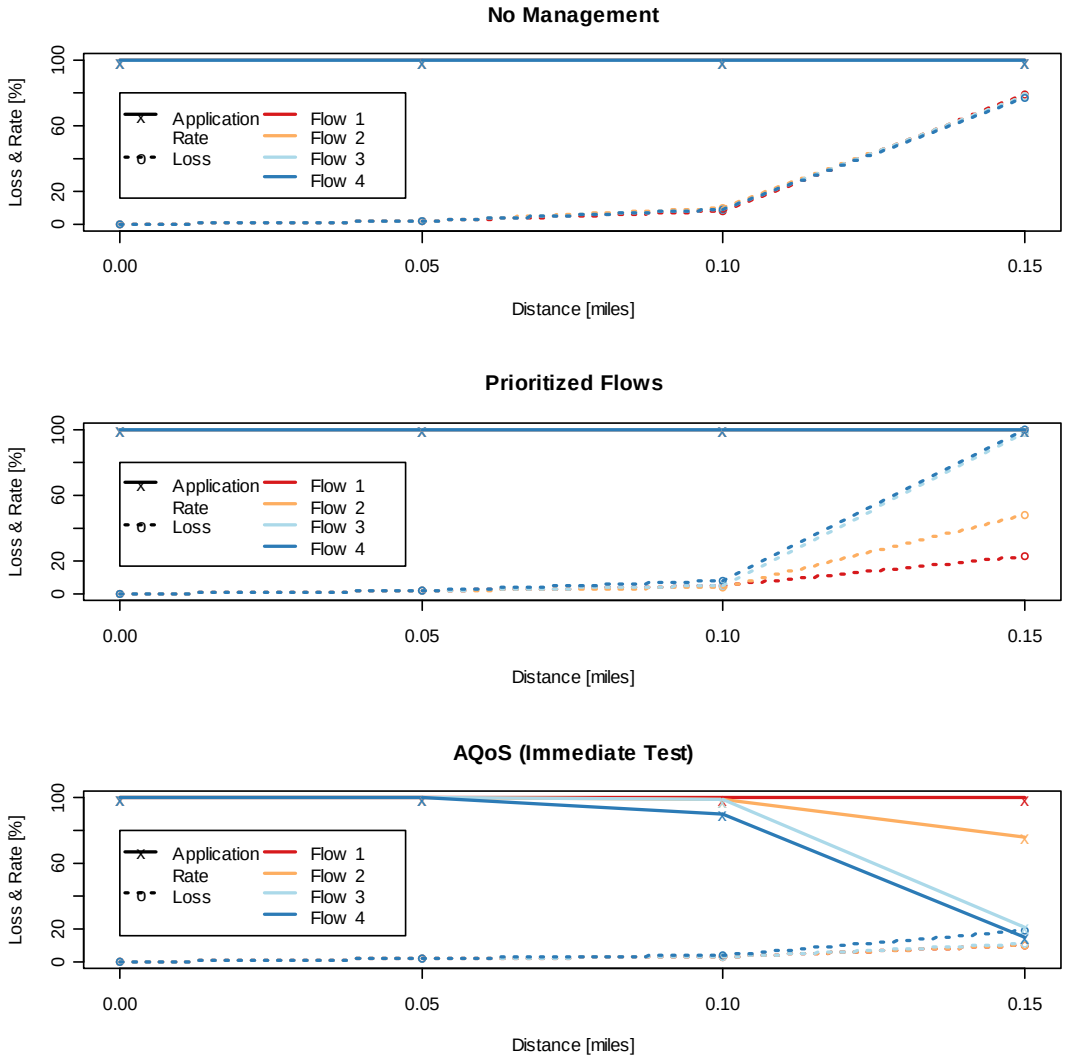

Figure 10: Iperf Discrete Tests in the Field

There is a noticeable difference in the loss patterns in Figure 9 and Figure 10 under Policies 2 and 3 (middle and bottom graphs, respectively, in the figures). In the laboratory, a flow starts losing data only if all lower priority flows lose all data or are highly quenched, whereas in the field noticeable data loss occurs on multiple flows during the same measurement.

The loss seen in Figure 9 is primarily due to drops in the queue introduced by the htb qdisc used in these experiments. That is, the software driver for the wireless radio on the router simply retrieved packets from the transmit queue and transmitted them at the programmed rate. Since this programmed rate was slower than the actual wireless link capacity, very few, if any, packets were actually lost "in the ether." They were simply dropped from the queue and never transmitted.

In this field experiment, the wireless signal weakened as the distance between the radios increased,[24] resulting in actual loss on the wireless link. To compensate, we believe the wireless radios switch their data rate mode.[25] As such, loss in the field has two possible causes: (1) loss on the wireless link and (2) drops resulting from the use of the htb qdisc. Our measurements were not sensitive to this distinction because we detect loss at the application level.

Undersubscription of the network was primarily between the base station and marker A. with little effect noticeable at marker B (0.10 miles) under all three policies. But when the distance increases to marker C, the network is clearly oversubscribed with all Iperf applications, operating under Policy 1, experiencing over 80% loss. Also notice that Policy 2 and Policy 3 have almost opposite effects at this distance. Policy 2 only distributes loss among flows according to priority, whereas under Policy 3 quenching reduces bandwidth for the flows according to priority while keeping loss rates low for all flows. Even at 0.15 miles, the higher priority application flows *f*1 and *f*2 are able to make progress with tolerable packet loss at a data rate close to their preferred data rate.

4.4.4 Experiment 3—VFS in the Laboratory

This experiment was conducted in the same manner as Experiment 1 (Section 4.4.2), but with the VFS application. Also, as in Experiment 1, the lines that connect the data points here only highlight the related points for each of the flows represented in the graph and are not intended to be predictions for intermediate rates.

The top graph in Figure 11 shows that under Policy 1 the video streams cumulatively oversubscribe the wireless link when the radios are set to any mode less than 54Mbit/s (see discussion in Section 4.1). Compared to the corresponding Iperf application graph in Experiment 1, loss rates increase faster initially when the 802.11g data rate mode is reduced. This is due to the fact that we measure loss at the application level in terms of video frames (rather than datagrams).[26] For the first two graphs, the preferred QoS level, level 4 in Table 8 on page 34, is constant where the resulting video frame is 10–13 times the size of the MTU on the wireless link. This means that for each successfully transmitted video frame, a sequence of 10–13 IP fragments must be successfully received. The loss rates of fragments and video frames are related. If we assume for simplicity that fragment losses occurs independent of each other, then

$$loss_{video} = 1 - \left(1 - loss_{fragment}\right)^{N},$$

where *N* is the number of fragments per video frame.

The middle graph in Figure 11 shows that Policy 2 has an effect on the VFS application flows similar to that seen on the Iperf application flows.

---

[24] This phenomenon was directly observed as the reported signal strength decreased with distance.

[25] In this version of the firmware (version r2690), the automatic rate change was not reported as expected. Prior versions of the firmware did report lower data rates. We do not yet know if this is simply a reporting error and will investigate it at a later date. As an unconfirmed fact, this has no bearing on our results since we know the signal as well as network capacity are diminished over distance.

[26] For the Iperf application, a datagram is less than the MTU. In the VFS application, a video frame is greater than the MTU and therefore is composed of many datagrams. Loss of even one of those composite datagrams means the entire video frame is lost.

The bottom graph of the same figure again shows that loss is reduced during diminished network capacity and that high-priority flows are protected from this resource constraint. There is an anomaly for the 54Mbit/s mode in that quenching occurs although there should be sufficient bandwidth available for all video streams. This quenching is likely due to some uncontrolled environmental influence.

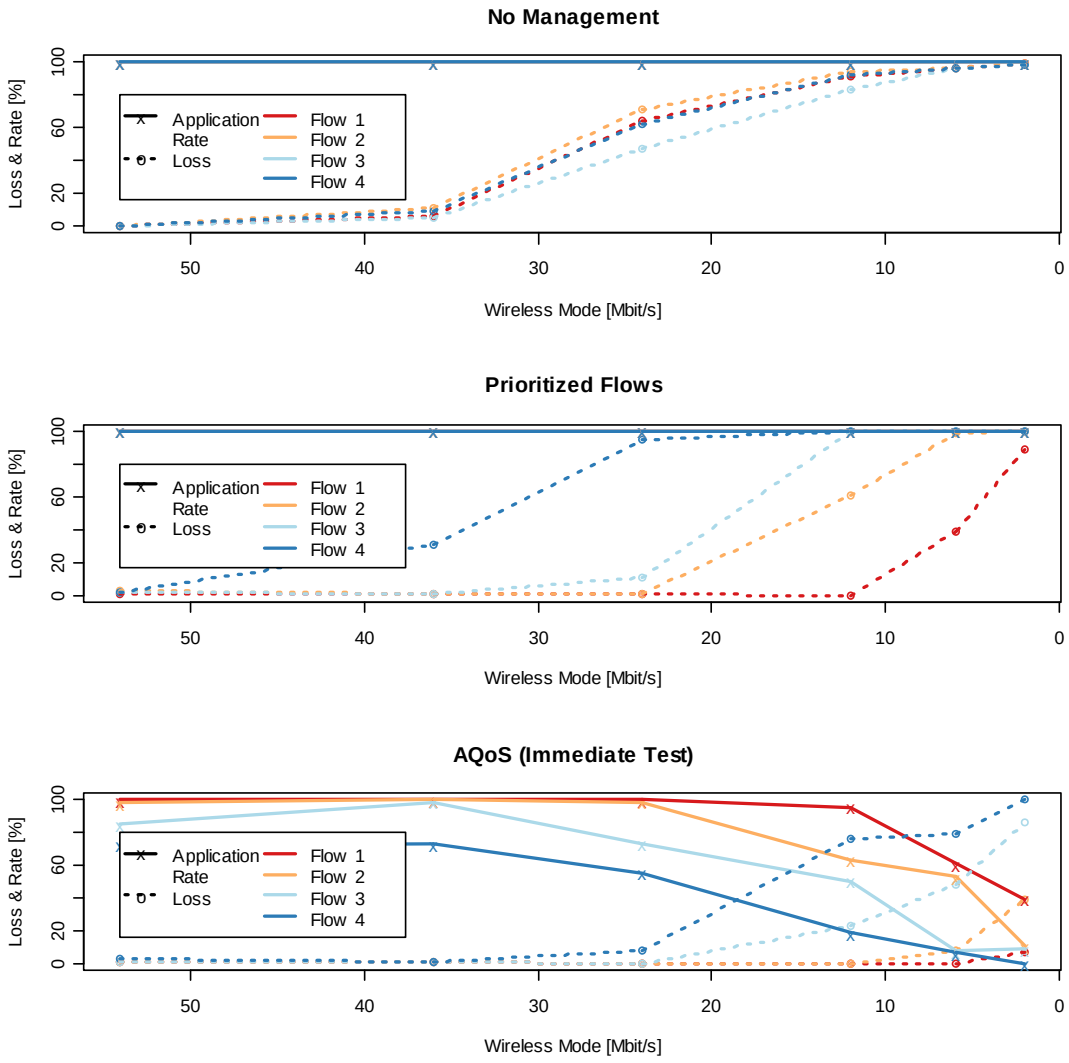

Figure 11: VFS Stationary Tests in the Laboratory

4.4.5 Experiment 4—VFS in the Field

This experiment was conducted in the same manner as Experiment 2 (Section 4.4.3), but with a reduction in duration from 10 minutes to 3 minutes. Note that this experiment was conducted on a different day than Experiment 2, so the results cannot be compared directly. For example, at 0.05 miles in Experiment 2, we saw an Iperf loss of up to 5%. We would then expect a video frame loss rate between 40% ($1-(1-0.05)^{10} \approx 0.4$) and 50% ($1-(1-0.05)^{13} \approx 0.5$), but we measured a loss of more than 60%. The graphs in Figure 12 show the experiment results.

The top graph shows how video frame loss increases, as expected, with distance under Policy 1.

Policy 2 led to a significant improvement of the loss rates for high-priority flows at medium distances over Policy 1. Furthermore, Policy 3 improves the loss rates for the lower priority flows at these greater distances while still protecting the highest priority VFS application flow, *f*1.

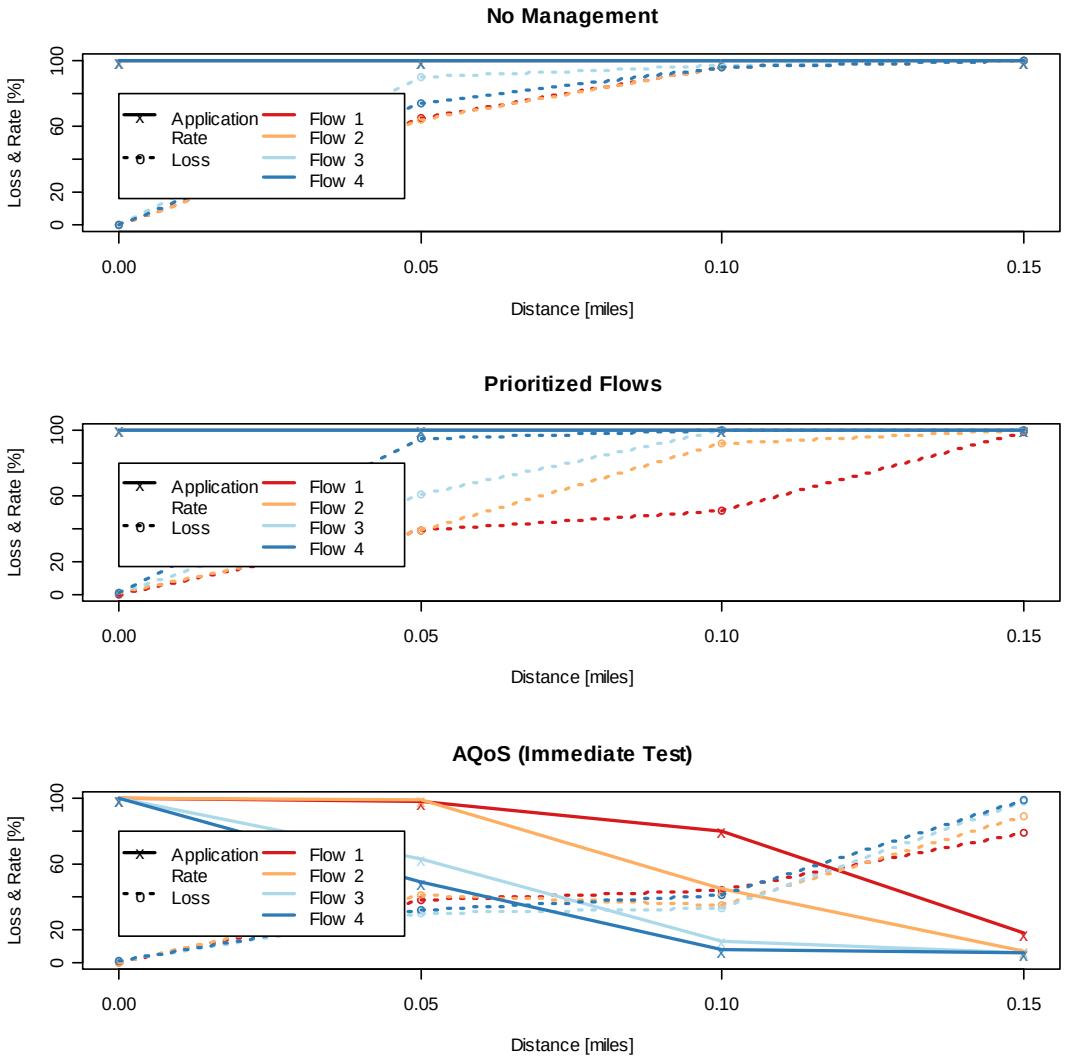

Figure 12 VFS Discrete Tests in the Field

When quantitatively comparing the results for the VFS application to those for Iperf, we see that AQoS shows a somewhat unexpected picture of the loss rates in that all four flows experience similar data loss. This observation can be explained by the different effects of quenching on the parameters of the application data streams:

- For Iperf, quenching changes only the rate at which data packets are sent out by the Iperf server.
- For the VFS, quenching changes the size of video frames (both JPEG compression and resolution), but the frame rate remains constant.

The effect with VFS is that fewer IP fragments are sent (as the video frame size decreases, approaching the MTU), which reduces even more the likelihood that an entire video frame will be lost due to the loss of a single IP fragment. As a result, higher priority, unquenched application flows can experience the same loss rate as lower priority, quenched application flows.

4.4.6 Experiment 5—Iperf Field Stress Test

The previous experiments achieved oversubscription by directly or indirectly reducing the available bandwidth on the wireless link. In this experiment, we increased the Iperf application data rate while keeping the available wireless bandwidth as constant as possible. We use the Iperf application to generate data streams at three times (15Mbit/s) and six times (30Mbit/s) above the cumulative of the preferred application rates in Table 5 on page 32. This experiment was conducted for three minutes under each policy for those data rates. For this experiment, the transmitter and receiver were positioned 0.10 miles apart (from the base station to marker B in Figure 3 on page 29).

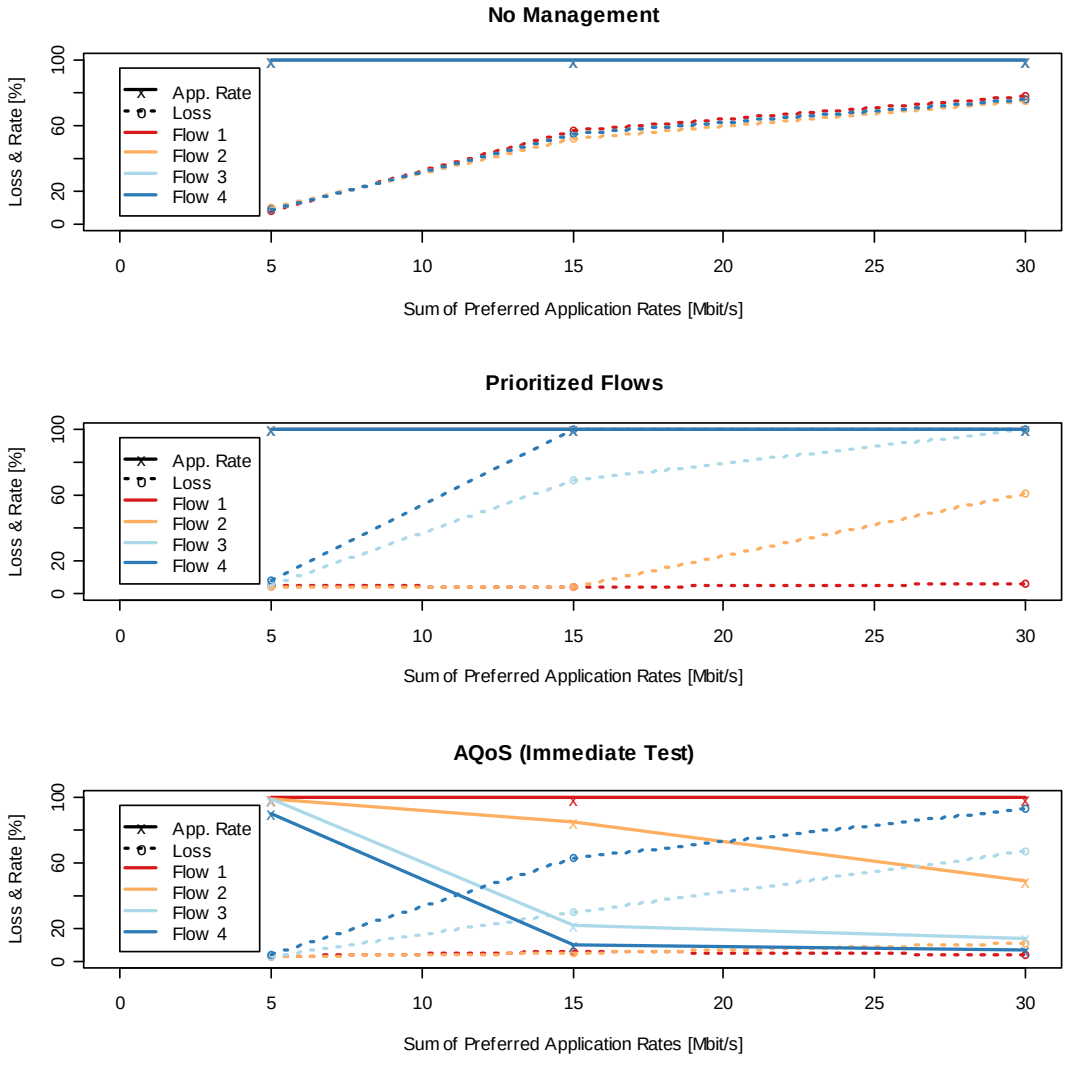

Figure 13: Iperf Stationary Tests in the Field

The graphs in Figure 13 show the expected behavior. There is quite a bit of similarity between the pattern of loss (and percentage of Application Data Rate change with Policy 3) seen here and in Experiment 1: drops due to filled queues rather than loss on the wireless link. This similarity can be explained by the phenomenon seen in the lab when the 802.11g data rate mode was fixed and the application data rate was higher than that mode. Packets were transmitted from the queue at the current data rate mode for the given distance and environmental conditions, and the bulk of those packets actually lost were simply never transmitted, but just dropped from the queue.

The graph for the No Management policy shows that a maximum data rate of approximately 7.5Mbit/s can be transmitted successfully over the wireless link and all other data is lost, leading to a loss rate of ~50% for 15Mbit/s and ~75% for 30Mbit/s. Note that we did not measure at 7.5Mbit/s and the dashed lines do not accurately approximate the loss there. Under Policy 2 data loss is pushed to the low-priority flows first, which allows the highest priority flow to maintain its preferred data rates (0.5Mbit/s, 1.5Mbit/s, and 3Mbit/s) with little loss. While under Policy 3, Iperf application data rates are reduced according to reverse priority ordering. The effect for Iperf is to trade bandwidth for limited data loss. As such, not only is the highest priority flow protected but the second highest priority flow can also progress with tolerable packet loss.

4.4.7 Experiment 6—VFS Continuous Field Test

Undersubscription and oversubscription were achieved in this experiment by constantly moving the wireless radios relative to each other. We did this to appropriately demonstrate the effects of AQoS compared to the other traffic management policies when faced with dynamic, real-time, variations in bandwidth caused by interference and mobility in the mobile wireless environment.

As noted earlier (see Figure 5 on page 31), the server machine was placed in a vehicle and driven through the course as laid out in Figure 3 on page 29. Starting at the base station, the vehicle travelled at five miles per hour past markers A, B, C, and D, was turned at the east turning point to return past the base station and markers E and F, was turned then at the west turning point and was returned to the base station. The control variable (X-axis) in this experiment is the time the vehicle has traveled. The duration of this experiment given the fixed distance and speed was just over 9 minutes.[27]

The top graph in Figure 14 shows how, initially, the wireless connection degraded and then was lost under Policy 1 when the vehicle moved too far away (150s – 250s). After the vehicle was turned around, connectivity improved until it passed the base station (~370s). After that, connectivity degraded again until the second turnaround point (~470s) and then improved again until the vehicle stopped near the base station at the end of the course.

As in other experiments, Policy 2 and Policy 3 were effective at shifting the negative effects of diminished network capacity. Conversely, the graphs in Figure 14 for Policy 2 and 3 show that the higher priority flows were able to use bandwidth, with increasing capacity and improving loss profiles, once connectivity between the radios was re-established after the vehicle entered wireless signal range.

---

[27] Rate was maintained to the best of the driver's ability.

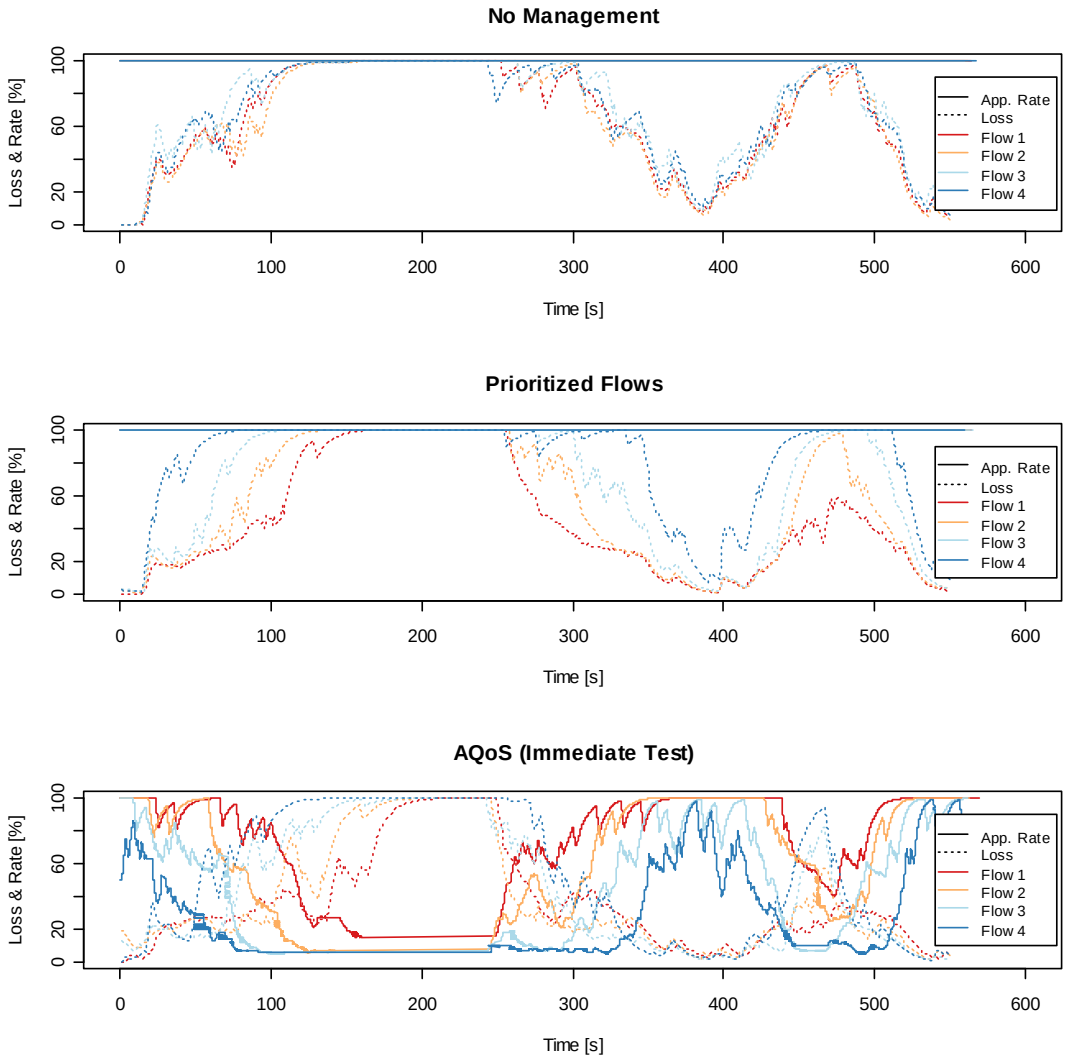

Figure 14: VFS Continuous Tests in the Field

The graphs in Figure 15 (following) show data loss rates under all three traffic management policies in the continuous field test. For all flows, loss rates are lowest for Policy 3, AQoS. Policy 3 reduces the number of sent data packets, thus sending only data with a high likelihood of reaching its destination. Policy 2 (Prioritized Flows) distributes data loss differently. It achieves reduced data loss only for the two highest priority flows and increases it for the lowest priority one.

The *All Flows* graph in Figure 15 shows the loss for the sum of all four flows. We see again that Policy 3 has the least overall loss. Policy 2 exhibits less overall loss than Policy 1(No Management). This difference is that a video frame consists of multiple fragments and loss is shifted away from the higher priority flows.

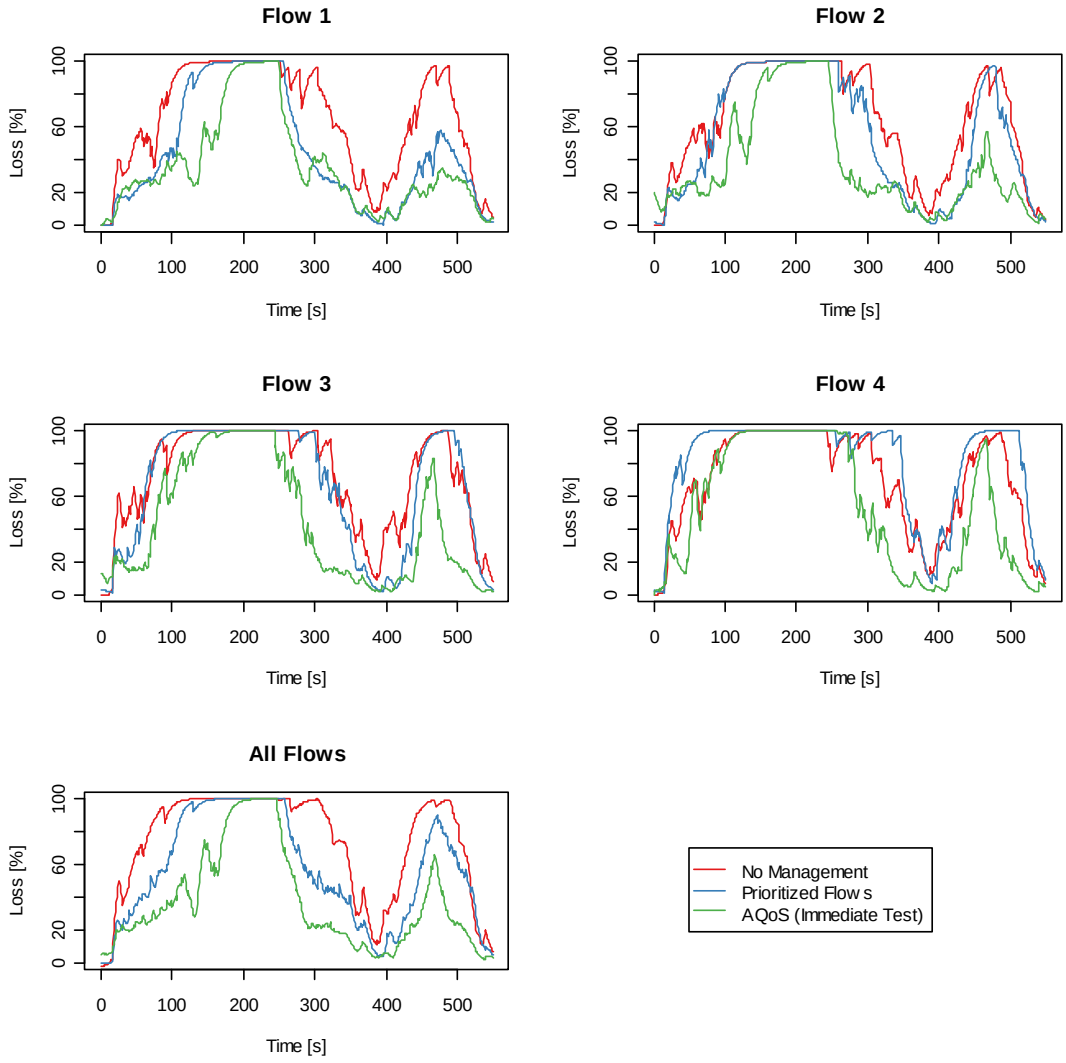

Figure 15: VFS Cumulative Loss Resulting from Continuous Tests in the Field

4.5 Summary Analysis of AQoS

In summary, the experiments have confirmed our hypotheses:

- Under Policy 1, No Management, all flows are impacted in the same way by data loss that occurs when the required bandwidth surpasses the available bandwidth on the wireless link (oversubscription).
- Under Policy 2, Prioritized Flows, data loss is shifted to low-priority flows, while high-priority flows continue to progress with tolerable packet loss if the wireless link becomes more and more oversubscribed.
- Under Policy 3, AQoS with Immediate Test, data loss rates are lowest as a tradeoff against lower data rates. The impact of quenching is most pronounced when the wireless link is weak. This effect is shown in Experiments 2 and 4 results, when the distance between radios increases to a point where Policy 1 leads to almost complete loss of data (>80%) for all flows. The impact of Policy 3 (AQoS) on loss and bandwidth clearly corresponds to the

priority of the flows, with high-priority flows being able to progress with tolerable packet loss in the face of diminishing network capacity.

5 Future Work

This section describes several topics for future work, including architecture, use of Q-RAM for resource allocation, multi-hop wireless networks, multiple traffic management policies, protocol considerations, route management, and device management. The variety of topics reflects the complexities and challenges of the domain.

5.1 Architecture

The architecture of any system warrants careful attention. For the ad hoc wireless domain the question of relevance is as follows: How does one construct and analyze a software architecture that accounts for the dynamic effects of the environment? For example, there may be a need for architectural elements for power management in battery-operated devices, where loss of power can have implications for network performance, such as connectivity of devices.[28] Another example is how to architecturally address route management when the route may change over short intervals of time. These examples suggest that the dynamic effects must be architecturally represented and analyzed in order to gain confidence that desired QoS characteristics can be achieved in dynamically changing environments.

5.2 QoS Resource Allocation Model (Q-RAM)

A key concern in this work is the ability to allocate resources in the presence of a dynamic environment such that those allocations are as optimal as possible. To address this issue, we propose exploring a utility-based optimization approach called Q-RAM [Lee 1999]. *Utility* is a number representing the benefit to the user of a particular QoS level for a chosen set of characteristics. For example, the utility for video QoS might include characteristics of frame rate, image resolution, and compression factor. Q-RAM has demonstrated success in heavily loaded situations [Lee 1999, Hoover 2001, and Hansen 2001] and is thus a candidate for consideration in this work.

We believe that Q-RAM can apply to the problem considered in this report as an approach to address concerns of resource adaptation in the face of dynamic effects. Q-RAM is an optimization heuristic for allocating resources to tasks in a way that optimizes utility. It is based on the Kuhn-Tucker condition [Kuhn 1951] that states that the marginal utility (i.e., the rate of change of utility with respect to resource) is equal across all tasks at the optimal point. The essential idea is that as network resources change, one may iteratively seek to optimize utility in the face of that change. For example, using the queue length for a flow as an indication of resource availability shortfalls, Q-RAM-based resource reallocation can be triggered when the queue length exceeds a specified threshold. Stated differently, a Q-RAM approach provides a *policy* for resource allocation that can provide near-optimal behavior of network nodes for QoS management.

[28] Consider the following example where a battery in a wireless device is treated as a *managed device*. We assume the existence of a data model for a battery supply and events that can be raised. In a situation where the battery power goes below a threshold value, an event will be raised. The existence of the event is then distributed to those components that have requested knowledge of the event. At this point, there must be some behaviors undertaken by other nodes to assure that network topology, and QoS provided over that topology, are maintained in accordance with application needs.

Several interesting challenges to the use of a Q-RAM approach (or any other optimization scheme for that matter) must be addressed relative to its use in a dynamic environment. From an implementation perspective, there are two notable concerns. First, a key parameter is the ratio of the time to compute the optimization of resource allocation to the time over which resource availability changes: it is necessary for the allocation scheme to be able to keep pace with the dynamic effects. Our preliminary estimate indicates that a distributed Q-RAM algorithm would be very efficient, thereby negating the concern. A second primary concern is the need for a *distributed* optimization approach that requires knowledge of what information must be shared as well as of how—and when—it is shared. Again, we believe that an approach based on Q-RAM would require little bandwidth to maintain an optimal solution. Thus, we believe the computational efficiency of the algorithm is appealing as providing a viable solution approach.

5.3 Multi-Hop Mesh Networks

Our experiments were performed using a single-hop wireless network (consisting of two wireless radios) and a model problem. In order to understand problems of large-scale wireless networks, the scope of component interactions must be broadened to a fully distributed, multi-hop wireless mesh network. Enlarging the scope will require consideration of how components interact in that broader context. For example, the role of when, and to which components, a quench message is distributed may be different when multiple nodes compose a path. This complexity is further exacerbated due to the fact that a route may change over a relatively short time interval. Furthermore, arbitration of network resources in this multi-hop, end-to-end environment must also be considered.

5.4 Multiple Traffic Management Policies

During the course of this work only the *Immediate* quench test was used for traffic management. However, it was recognized that other quench tests can be useful (see Section 3.4.2.2) and can perform better in some cases. For example, by its nature the *AQoS with Immediate Test* policy (see Section 4.1) may overly respond to a spike in traffic for a flow. However, adverse effects of this over-response can be mitigated by applying an *AQoS with Over Time with Slope Test* policy. This illustrates the need, and value, of using multiple traffic management policies. Comparing the effects of multiple traffic management policies will be part of future work. Furthermore, the ability to apply multiple policies to different queues is also of interest. Under this situation, the interplay of traffic management policies applied to different queues provides opportunities that may prove valuable.

5.5 Protocol Considerations

The transport protocol chosen for this work was UDP. The rationale for this choice was to avoid concerns surrounding TCP congestion control that may prove inefficient in a wireless ad hoc mesh network [Plakosh 2010]. Flow control in TCP is based on joint considerations of both sender and receiver. In brief, a sender has a defined window size and it will not transmit more segments than can be accommodated. The receiver may invoke a flow control approach to inform the sender of the desired window size such that transmission from the sender to the receiver will likely be successful. It may, however, be the case that applying different traffic management policies can help to mitigate the congestion avoidance concerns associated with TCP [Kliazovich 2006],

including the end-to-end approach that may conflict with a dynamic environment where a route can change over a small interval of time.

Another question related to the choice of a transport protocol is the distribution of priorities. As is well known, neither UDP nor TCP supports a priority field in the header. Thus, some additional method must be used if one seeks to include priorities associated with flows (or messages that compose a flow). The approach taken in our experiments was to pre-assign priorities to port numbers, thereby avoiding representation of priority in a transport (or other) protocol data unit. The consequences of this choice, and other possible alternatives, warrant further examination.

5.6 Route Management in an Ad Hoc Wireless Network

Route management in a wireless network is known to be of concern due to changes in the network route path. In particular, a path can change for various reasons such as topological considerations, mobility of nodes, or available power (both reception and transmission) in a node. Route changes can occur frequently.

OLSR and B.A.T.M.A.N. make different assumptions regarding the approach to route management. For example, OLSR assumes that there is a global view of network connectivity, while B.A.T.M.A.N. focuses on only nearest neighbors. In either case, QoS concerns need to be addressed. One approach might be to distribute QoS characteristics (such as the queue size, bandwidth, or available power) among routers. Following the approach taken in this work, we could include a software element that is a QoS Management Agent for Route Management (QOSMA.RM) that interacts with other QoSMA.RMs in a multi-hop wireless mesh network.

In the current approach, we used OLSR as the routing scheme. However, B.A.T.M.A.N. offers the significant advantage of being based on local scope. We are especially interested in the ability to distribute QoS metrics (such as queue size) as well as custom policies that may use those metrics.

5.7 Device Management

In general, device management includes elements related to (1) a data model of a device, (2) events that can be raised by the device, (3) behaviors that may be initiated by a management agent, and (4) a protocol to support communication between management agents. In this report, we have treated the VFS as a managed device but without explicitly considering the above elements.

There are a number of organizations that are concerned with development of standards for device management including

- the **Universal Plug and Play (UPnP) Forum**,[29] which focuses on devices related to consumer electronics
- the **Distributed Management Task Force (DMTF)**,[30] which focuses on management of information technology

[29] For more information, visit http://www.upnp.org/.

[30] For more information, visit http://www.dmtf.org/.

- the **Open Mobile Alliance (OMA)**,[31] which focuses on delivery of interoperable services for the mobile environment

A limited examination of specifications provided by these organizations indicates that while much work has been done, there has not been sufficient treatment of QoS considerations to warrant application to a wireless network. Thus, we conclude that the role of QoS in the context of device management needs further consideration.

There is also the larger role of device management in some network contexts that must be considered. For example, suppose a battery-operated device is configured to raise an event if its power level drops below a previously specified value. Given the presence of the event, what behavior of other network nodes is expected? Are the behaviors preconfigured prior to deployment, or determined and distributed during operational use? These two questions are general in nature and pose a formidable range of alternatives that comprise a large solution space for device management.

[31] For more information, visit http://www.openmobile alliance.org/.

6 Conclusions

The objectives of the experiments discussed in this report were to investigate the viability of AQoS, where the networking infrastructure reacts and mission applications adapt to dynamic changes in available network capacity. As a result of using AQoS, it is intended that mission applications operating in mobile, ad hoc, wireless networks can continue to operate at levels that serve mission goals given the overall state of demand and capacity of the network.

The AQoS approach does not need to predict bandwidth capacity; it simply integrates prioritization of flows with network congestion feedback to applications. AQoS is capable of monitoring priority queues deployed in the networking infrastructure as an "early warning system" for network congestion. These observations are used by the network infrastructure to notify mission-critical applications of such conditions. Applications can use this information to adapt to changing network service levels while continuing to achieve mission goals.

Based on experiments to date, in undersubscription network scenarios, demonstrated performance of AQoS is no worse than that of the other traffic management policies defined for our investigation. More importantly, for oversubscription scenarios, AQoS allows applications to adapt their bandwidth demand in a controlled manner in response to available network bandwidth, thereby avoiding unpredictable packet loss due to congestion (i.e., oversubscription).

Although the assumptions and constraints established for our initial experiments were limiting, our assessment of the AQoS approach is encouraging. Achieving continued and usable service for high-priority applications in the face of dynamically changing and diminishing (or conversely, increasing) network capacity without the need to reserve network bandwidth or to know, *a priori*, available network bandwidth is a step forward.

In future work, we intend to address many of the noted limiting factors in our experiments. Specifically, we plan to investigate utility-based degradation of service in response to resource shortfalls. In our experiments, the low-priority task could be completely starved in order to meet the full demands of the high-priority task. Under a utility-based approach the notion of priority is generalized so that, a low-priority task could be allowed to operate at a minimal level in exchange for a slight degradation of a high-priority task. These allocation decisions lead to increased system-wide benefit. Further, the wireless network in our experiment had only two nodes. Ultimately, AQoS will have to scale; it must operate in multi-node, wireless networks such as mobile ad hoc wireless mesh networks, which are self-configuring and self-healing.

Changes to the networking infrastructure were necessary to support AQoS. These changes included modifications to the open source Open-Mesh OM1P Professional Mini Router, predominately to modify the router's default configuration and to add our QOSMA.FC software. Changes were also made to the existing open source Iperf application to support messages sent by the QOSMA.FC. In addition, we developed our own implementation for VFS. Working with these applications allowed us to begin to consider longer term questions, such as "Can applications be built that provide requisite flexibility for edge users, yet make efficient use of ad hoc wireless network resources?"

Appendix A Commands for Traffic Control and HTB Qdisc

The following suite of commands shown in Figure 16 and Figure 17 set up the queue structure as discussed in Figure 2 on page 22.

```
QDEV=wifi0;export QDEV
tc qdisc add dev ${QDEV} handle 2:0 root htb default 6 r2q 10 # SAH
tc class add dev ${QDEV} parent 2:0 classid 2:1 htb \
            rate 7077888bps ceil 7077888bps burst 10486 cburst 5243 prio 1
tc class add dev ${QDEV} parent 2:1 classid 2:2 htb \
            rate 1bps ceil 7077888bps burst 10486 cburst 5243 prio 2
tc class add dev ${QDEV} parent 2:1 classid 2:3 htb \
            rate 1bps ceil 7077888bps burst 10486 cburst 5243 prio 3
tc class add dev ${QDEV} parent 2:1 classid 2:4 htb \
            rate 1bps ceil 7077888bps burst 10486 cburst 5243 prio 4
tc class add dev ${QDEV} parent 2:1 classid 2:5 htb \
            rate 1bps ceil 7077888bps burst 10486 cburst 5243 prio 5
tc class add dev ${QDEV} parent 2:1 classid 2:6 htb \
            rate 65536bps ceil 65536bps burst 10486 cburst 5243 prio 8
tc qdisc add dev ${QDEV} handle 3:0 parent 2:2 sfq
tc qdisc add dev ${QDEV} handle 4:0 parent 2:3 sfq
tc qdisc add dev ${QDEV} handle 5:0 parent 2:4 sfq
tc qdisc add dev ${QDEV} handle 6:0 parent 2:5 sfq
tc qdisc add dev ${QDEV} handle 7:0 parent 2:6 sfq
```

Figure 16: tc Commands to Establish the htb and sfq Queues

```
QDEV=wifi0;export QDEV
PROTO=all;export PROTO
tc filter add dev ${QDEV} parent 2:0 \
            protocol ${PROTO} prio 1 handle 2 fw classid 2:2
tc filter add dev ${QDEV} parent 2:0 \
            protocol ${PROTO} prio 2 handle 3 fw classid 2:3
tc filter add dev ${QDEV} parent 2:0 \
            protocol ${PROTO} prio 3 handle 4 fw classid 2:4
tc filter add dev ${QDEV} parent 2:0 \
            protocol ${PROTO} prio 4 handle 5 fw classid 2:5
tc filter add dev ${QDEV} parent 2:0 \
            protocol ${PROTO} prio 5 handle 6 fw classid 2:6
```

Figure 17: tc Commands to Set Up the fw Filters

The filters created in Figure 17 associate different protocol stream marked in Figure 18 to htb queues defined in Figure 16.

```
iptables -t mangle -A PREROUTING -j CONNMARK --restore-mark
iptables -t mangle -A PREROUTING -m mark ! --mark 0 -m mark ! --mark 1024 -j
ACCEPT
iptables -t mangle -A PREROUTING -p udp --dport 10001 -j MARK --set-mark 2
iptables -t mangle -A PREROUTING -p udp --dport 10002 -j MARK --set-mark 3
iptables -t mangle -A PREROUTING -p udp --dport 10003 -j MARK --set-mark 4
iptables -t mangle -A PREROUTING -p udp --dport 10004 -j MARK --set-mark 5
iptables -t mangle -A PREROUTING -j CONNMARK --save-mark
#
# now setup the accept rules in the filter table for these marks
# we are setting in the mangle table.
#
iptables -t filter -I ndsNET 4 -m mark --mark 0x2 -j ACCEPT
iptables -t filter -I ndsNET 4 -m mark --mark 0x3 -j ACCEPT
iptables -t filter -I ndsNET 4 -m mark --mark 0x4 -j ACCEPT
iptables -t filter -I ndsNET 4 -m mark --mark 0x5 -j ACCEPT
```

Figure 18: iptables Commands to Mark Packets for fw Classification

Appendix B Linux Kernel Modifications

Early tests in the lab were able to demonstrate limited success with AQoS in the context of our model problem. The failures experienced were caused by the inability to monitor queue lengths in oversubscribed situations. In such situations, queue lengths simply indicated that there was no oversubscription occurring when it was clear that the link was saturated (due to observed packet loss and the fact that Iperf was set to transmit well above the link's capacity... 1Gbit/s over a 54Mbit/s link!).

The cause was actually an architectural violation that appears to have been, in someone's belief, an "optimization." On the router, the physical radio device driver (i.e., `wifi0`) handles the hardware direct memory access (DMA) queue for the radio's transmit buffer. The radio's access point is a virtual "child" device (i.e., `ath0`) to the physical interface. The failed optimization, added to the virtual device driver, `ath0`, was to inspect the state of the hardware DMA transmit buffer and if stopped simply drop/free the packet.

The violation was traced to a single line of code. That line of code was removed from the madwifi driver (revision r3314) in `ieee80211_output.c` (a module for the Linux kernel used for the wireless radio hardware used in the OM1P router).

```
1  --- a/net80211/ieee80211_output.c
2  +++ b/net80211/ieee80211_output.c
3  @@ -324,9 +324,10 @@ void ieee80211_parent_queue_xmit(struct
4           /* Dispatch the packet to the parent device */
5           skb->dev = vap->iv_ic->ic_dev;
6
7  -        if (dev_queue_xmit(skb) == NET_XMIT_DROP)
8  +        if (netif_queue_stopped(skb->dev))
9  +                ieee80211_dev_kfree_skb(&skb);
10 +        else if (dev_queue_xmit(skb) == NET_XMIT_DROP)
11                  vap->iv_devstats.tx_dropped++;
12 -
13 }
```

Figure 19: 372-queue_vif.patch

The original line of code, which was removed by the application of the patch, line 7 of Figure 19 (removal denoted with a minus sign), simply passes the packet to be transmitted to the physical parent device driver (via `dev_queue_xmit()`). The test introduced by the patch, line 8 of Figure 19 (addition denoted with a plus sign), is not supposed to be done in the "child" device driver, but is the responsibility of the physical "parent" device, `wifi0`.

The result of replacing line 7 with lines 8-10 in Figure 19 is that the packet to be sent **would not be queued** for later transmission and **silently dropped** if the DMA transmit buffer were full even if there were space available in the software queue, which in this case was the htb qdisc.

The original line of code was the behavior needed for AQoS and the QOSMA.FC.

Bibliography/References

URLs are valid as of the publication date of this document.

[Abolhasan 2009]
Abolhasan, M., Hagelstein, B., & Wang, J.C.-P. "Real-World Performance of Current Proactive Multi-Hop Mesh Protocols," 44-47. *Proceedings of the 15th IEEE Asia-Pacific Conference on Communications (APCC'09).* Shanghai, China, October 2009. IEEE, 2009.

[Batman 2010]
Open Mesh. *Welcome to Open Mesh: B.A.T.M.A.N*. http://www.open-mesh.org (2010).

[Blake 1998]
Blake, S., Black, D., Carlson, M., & Davies, E. *An Architecture for Differentiated Services* (RFC 2475). Network Working Group, December 1998. http://rfc-ref.org/RFC-TEXTS/2475/index.html

[Bose 2008]
Bose, P., Zimdars, A., Quilling, M., Slavin, V., & ElBatt, T. "Network Utility Maximization-Based Mobile Ad Hoc Networking: A Reality Check," 1-7. *Proceedings of the IEEE Military Communications Conference (MILCOM 2008).* San Diego, CA (USA), November 2008. IEEE, 2008.

[Braden 1994]
Braden, R., Clark, D., & Shenker, S. *Integrated Services in the Internet Architecture: an Overview* (RFC 1633). Network Working Group, June 1994. http://rfc-ref.org/RFC-TEXTS/1633/index.html

[Braden 1997]
Braden, R., Zhang, L., Berson, S., Herzog, S., & Jamin, S. *Resource ReSerVation Protocol (RSVP) Version 1 Functional Specification* (RFC 2205). Network Working Group, September 1997. http://rfc-ref.org/RFC-TEXTS/2205/index.html

[CBO 2003]
Congressional Budget Office. *The Army's Bandwidth Bottleneck*. The Congress of the United States, 2003.

[Chen 2004]
Chen, D., Pramod, K., & Varaiya, P. "QoS Support in Wirelss Sensor Networks: A Survey," 227-233. *Proceedings of the International Conference on Wireless Networks (ICWN 2004).* Las Vegas, NV (USA), June 2004. CSREA Press, 2004.

[Chen 2005]
Chen, L. & Heinzelman,W. "QoS-Aware Routing Based on Bandwidth Estimation for Mobile Ad Hoc Networks." *IEEE Journal on Selected Areas in Communications - Special Issue on Wireless Ad Hoc Networks 23*, 3 (March 2005): 561-572.

[Chen 2007]
Chen, L. & Heinzelman, W. "A Survey of Routing Protocols that Support QoS in Mobile Ad Hoc Networks." *IEEE Network 21*, 6 *(*December 2007): 30-38.

[Chung 2002]
Chung, Joe & Claypool, Mark. "Rate-Based Active Queue Management with Priority Classes for Better Video Transmission." *Proceedings of the 7th International Symposium on Computers and Communication (ISCC'02).* Taormina-Giardini Naxos, Italy, July 2002. IEEE, 2002.

[Clausen 2003]
Clausen, T. & Jacquet, P. Optimized Link State Routing Protocol (OLSR), Internet Request for Comment RFC 3626, 2003.

[Demers 1990]
Demers, A., Keshav, S., & Shenker, S. "Analysis and Simulation of a Fair Queueing Algorithm." *Journal of Internetworking Research and Experience 1*,1 (October 1990): 3-26.

[Devera 2003]
Devera, M. *Hierarchical Token Bucket*. http://luxik.cdi.cz/~devik/qos/htb/ (2003).

[Ergen 2003]
Ergen, M., Coleri, S., & Varaiya, P. "QoS Aware Adaptive Resource Allocation Techniques for Fair Scheduling in OFDMA Based Broadband Wireless Access," 362-370. *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking (MobiHoc 2003).* Annapolis, MD (USA), June 2003. ACM, 2003.

[Felemban 2005]
Felemban, E., Lee, C-G., Ekici, E., Boder, R., & Vural, S. "Probabilistic QoS Guarantee in Reliability and Timeliness Domains in Wireless Sensor Networks," 2646-2657 vol. 4. *Proceedings of the 24th IEEE International Conference on Computer Communications (INFOCOM 2005).* Miami, FL (USA), March 2005. IEEE, 2005.

[Feng 2001]
Feng, Wu-cheng, Shin, Kang G., Kandlur, Dilip, & Saha, Debanjan. "The BLUE Active Queue Management Algorithms." *IEEE/ACM Transactions on Networking.10*, 4 (August 2002): 513-528.

[Ge 2003]
Ge, Y., Kunz, T., & Lamont, L. "Quality of Service Routing in Ad-Hoc Networks using OLSR." *Proceedings of the 36th Hawaii International Conferences on Systems Science.* Big Island, HI (USA), January 2003. IEEE, 2003.

[Han 2007]
Han, Z., Liu, X., Wang, Z., & Liu, K. J. R. "Delay Sensitive Scheduling Schemes for Heterogeneous QoS over Wireless Networks." *IEEE Transactions on Wireless Communications 6*, 2 (February 2007): 423-428.

[Hansen 2001]
Hansen, J., Lehoczky, J., & Rajkumar, R. "Optimization of Quality of Service in Dynamic Systems," vol. 3, 30095b. *Proceedings of the 15th International Parallel and Distributed Real-Time Systems Symposium*. San Francisco, CA (USA), April 2001. IEEE, 2001.

[Hoover 2001]
Hoover, C., Hansen, J., Koopman, P. & Tamboli, S. "The Amaranth Framework: Policy-Based Quality of Service Management for High Assurance Computing." *International Journal of Reliability, Quality and Safety Engineering 8*, 4 (December 2001): 323-350.

[Hou 2009a]
Hou, I-H. & Kumar, P. R. "Admission Control and Scheduling for QoS Guarantees for Variable-Bit-Rate Applications," 175-184. *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc).* New Orleans, LA (USA), May 2009. ACM, 2009.

[Hou 2009b]
Hou, I-H., Borkar, V., & Kumar, P. R. "A Theory of QoS for Wireless," 486-494. *Proceedings of the 28th IEEE International Conference on Computer Communications (INFOCOM 2009).* Rio de Janeiro, Brazil, April 2009. IEEE, 2009.

[Huang 2004]
Hunag, L., Kumar, S., & Kuo, J. "Adaptive Resource Allocation for Multimedia QoS Management in Wireless Networks." *IEEE Transactions on Vehicular Technology 53*, 2 (March 2004): 547-558.

[Hwang 2010]
Hwang, I. S., Hwang, Bor-Jiunn, Chang, Pen-Ming, & Wang, Cheng-Yu. "QoS-Aware Active Queue Management for Multimedia Services over the Internet." *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS 2010)* (Lecture Notes in Engineering and Computer Science, Vol II). Hong Kong, March 2010. www.iaeng.org/publication/IMECS2010/IMECS2010_pp774-779.pdf

[Issariyakul 2008]
Issariyakul, T. & Hossain, E. *Introduction to Network Simulator NS2*. Springer, 2008. ISBN 978-0-387-71759-3

[Julian 2002]
Julian, D. D., Chiang, M., O'Neill, D., & Boyd. S. "QoS and Fairness Constrained Convex Optimization of Resource Allocation for Wireless Cellular and Ad Hoc Networks," 1-10. *Proceedings of the 21st IEEE International Conference on Computer Communications (INFOCOM 2002).* New York, NY (USA), June 2002. IEEE, 2002.

[Kelly 1998]
Kelly, F. P., Maulloo, A. K., & Tan, D. K. H., "Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability." *The Journal of Operational Research Society 49*, 3 (March 1998): 237-252.

[Klein 2008]
Klein, Mark, Plakosh, Daniel & Wallnau, Kurt. *Using the Vickrey-Clarke-Groves Auction Mechanism for Enhanced Bandwidth Allocation in Tactical Data* (CMU/SEI-2008-TR-004). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2008. http://www.sei.cmu.edu/library/abstracts/reports/08tr004.cfm

[Kliazovich 2006]
Kliazovich, D. & F. Granelli, Cross_Layer *Congestion Control in ad hoc Wireless Networks*, Ad Hoc Networks, vol. 4, pp. 687-708 (2006).

[Koo 2005]
Koo, Jahwan, Ahn, Seongjin, & Chung, Jimwook. "A Comparative Study of Queue, Delay, and Loss Characteristics of AQM schemes in QoS-Enabled Networks." *Journal of Computing and Informatics 23* (2004): 317-335.

[Kuhn 1951]
Kuhn, H. W. & Tucker, A. W. "Nonlinear programming," 481–492. *Proceedings of 2nd Berkeley Symposium on Mathematical Statistics and Probability*. University of California, Berkeley, July 31-August 12, 1950. University of California Press, 1951.

[Lee 1999]
Lee, C., Lehoczky, J., Rajkumar, R. & Hansen, J. "A Scalable Solution to the Multi-Resource QoS Problem," 315-326. *Proceedings of the 20th IEEE Real-Time Systems Symposium.* Phoenix, AZ (USA), December 1999. IEEE, 1999.

[Li 2001]
Li, J., Blake, C., De Couto, D., Lee, H., & Morris, R. "Capacity of Ad Hoc Wireless Networks," 61-69. *Proceedings of ACM SIGMOBILE 7th Annual International Conference on Mobile Computing (MobiCom 2001)*. Rome, Italy, July 2001. ACM, 2001.

[Liu 2006]
Liu, Q., Wang, X., & Georgios, G. "A Cross-Layer Scheduling Algorithm with QoS Support in Wireless Networks." *IEEE Transactions on Vehicular Technology 55*, 3 (May 2006): 839-847.

[Luo 2004]
Luo, H., Lu, S., Bharghavan, V., Cheng, V., & Zhong, G. "A Packet Scheduling Approach to QoS Support in Multiphop Wireless Networks." *ACM Journal of Mobile Networks and Applications (MONET) 9*, 3 (June 2004): 193-206.

[Manoj 2009]
Manoj, K., Parmanand, S. C. Sharma, & Singh, S. P. "Performance of QoS Parameter in Wireless Ad hoc Network (IEEE 802.11b)," 280-284. *Proceedings of the World Congress on Engineering and Computer Science (WCECS 2009)*. San Francisco, CA (USA), October 2009. IAENG, 2009.

[McKenney 1990]
McKenney, P. E. “Stochastic fairness queueing,” 733-740 vol. 2. *Proceedings of the 9th Annual Joint Conference of the IEEE Computer and Communication Societies, ‘The Multiple Facets of Integration’. (INFOCOM '90)*. San Francisco, CA (USA), June 1990. IEEE, 1990. DOI: 10.1109/INFCOM.1990.91316.

[Meissner 2002]
Meissner, A., Luckenbach, T., Risse, T., Kirste, T., & Kirchner, T., “Design Challenges for an Integrated Disaster Management Communication and Information System”, *Proceedings of Information System. (DIREN 2002, co-located with IEEE INFOCOM 2002)*, New York, New York (USA), June 2002, IEEE, 2002. DOI: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.5105

[Nagle 1987]
Nagle, J. “On Packet Switches with Infinite Storage.” *IEEE Transactions on Communications 35*, 4 (April 1987): 435–438.

[Nichols 1998]
Nichols, K., Blake, S., Baker, F., & Black, D. *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers* (RFC 2474). Network Working Group, December 1998. http://rfc-ref.org/RFC-TEXTS/2474/index.html

[Oh 2010]
Oh, S., Marfia, G., & Gerla, M. “MANET QoS Support without Reservations.” *Security and Communication Networks*. Wiley Interscience Online (www.interscience.wiley.com). DOI: 10.1002/sec.183 (2010)

[Perkins 2003]
Perkins, C. *Ad hoc On-Demand Distance Vector (AODV) Routing* (RFC 3561). Network Working Group, July 2003. http://rfc-ref.org/RFC-TEXTS/3561/index.html

[Plakosh 2008]
Plakosh, D., Klein, M., Moreno, G., & Wallnau, K. “Mechanism Design,” 23-33. *Results of SEI Independent Research and Development Projects* (CMU/SEI-2008-TR-025). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2008. http://www.sei.cmu.edu/library/abstracts/reports/08tr025.cfm

[Plakosh 2010]
Plakosh, D., Simanta, S., Morris, E., Anderson, W., & Seibel, J. “Web Services for Ad Hoc and Resource-Impoverished Environments.” *Proceedings of the 2010 Networking and Electronic Commerce Research Conference* (NAEC 2010). Riva del Garda, Italy, October 7-10, 2010.

[Polk 2006]
Polk, J., and S. Dhesikan. "A Resource Reservation Protocol (RSVP) Extension for the Reduction of Bandwidth of a Reservation Flow." *RFC 4495*. Network Working Group, May 2006.

[Proxim 2003]
Proxim Wireless Networks. *A Detailed Examination of the Environmental and Protocol Parameters that Affect 802.11g Network Performance*, Proxim Wireless Corporation, Milpitas, CA, June 2003.
http://www.proxim.com/learn/library/whitepapers/parameters_802.11g_performance.pdf

[Sanzgiri 2004]
Sanzgiri, K., Chakeres, I., & Belding-Royer, E. "Determining Intra-Flow Contention among Multihop Paths in Wireless Networks," 611-620. *Proceedings of 1st International Conference on Broadband Networks (BroadNets 2004).* San Jose, CA (USA), October 2004. IEEE, 2004.

[Shen 2008]
Shen, Q., Fang, X., Li, P., & Fang, Y. "Admission Control for Providing QoS in Wireless Mesh Networks," 2910-2914. *Proceedings of the International Conference on Communications ICC 2008)*. Beijing, China, May 2008. IEEE, 2008.

[Sinha 1999]
Sinha, P., Sivakumar, R., & Bharaghavan, V., "CEDAR: A Core-Extraction Distributed Ad hoc Routing Algorithm." *IEEE Journal on Selected Areas In Communication 17*, 8 (August 1999): 1454-1465.

[Tang 2005]
Tang, J., Guolian, X., & Weiyi, Z. "Interference-Aware Topology Control and QoS Routing in Multi-Channel Wireless Mesh Networks," 68-77. *ACM International Symposium on Mobile Ad Hoc Networking & Computing.* Urbana-Champaign, IL (USA), May 2005. ACM. 2005.

[Wikimedia 2010]
Wikimedia Foundation. *Motion JPEG. 2010*. http://en.wikipedia.org/wiki/Motion_JPEG (2010).

[Wilson 2005]
Wilson, Clay. *Network Centric Warfare: Background and Oversight Issues for Congress, March 18, 2005* (Congressional Research Service Report Number RL32411). Library of Congress, 2005. http:// www.au.af.mil/au/awc/awcgate/crs/rl32411.pdf

[Wilson 2007]
Wilson, C. *Network Centric Operations: Background and Oversight Issues for Congress, Updated March 15, 2007* (Congressional Research Service Report for Congress Number RL32411). Library of Congress, 2007. http://www.fas.org/sgp/crs/natsec/RL32411.pdf

[Xue 2003]
Xue, Q. & Ganz. A. "Ad Hoc QoS On-Demand Routing (AQOR) in Mobile Ad hoc Networks." *Journal on Parallel and Distributed Computing 62*, 2 (February 2003): 154-165.

[Zhang 1993]
Zhang, L., Deering, S., Estrin, D., Shenker, S., & Zapala, D. "RSVP: A New Resource ReSerVation Protocol." *IEEE Network 7*, 5 (September 1993): 8-18.

[Zhang 2005]
Zhang, Q., Wenwu, Z., & Zhang, Y-Q. “End-to-End QoS for Video Delivery Over Wireless Internet.” *Proceedings of the IEEE 93*, 1 (January 2005): 123-134.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	December 2010	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Adaptive Flow Control for Enabling Quality of Service in Tactical Ad Hoc Wireless Networks	FA8721-05-C-0003

6. AUTHOR(S)

Jeffrey Hansen, Scott Hissam, Craig B. Meyers, Ed Morris, Daniel Plakosh, Soumya Simanta, Lutz Wrage

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2010-TR-030

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116	ESC-TR-2010-030

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT	12B DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

13. ABSTRACT (MAXIMUM 200 WORDS)

Wireless networks for emergency responders and military personnel operating in tactical situations are often assembled without any preexisting infrastructure (i.e., ad hoc) and are subject to changing topology as nodes enter or leave service or move (i.e., are mobile) in the environment. These networks often have lower-than-optimal bandwidth and can see further bandwidth reductions due to disadvantageous topologies and other factors. In addition, needed applications must compete for possibly diminishing bandwidth. As a result, such networks are frequently oversubscribed: they cannot fully meet the quality of service (QoS) expectations of all applications.

This report provides an overview of approaches for satisfying QoS expectations in ad hoc wireless networks assembled to support high-criticality crisis and tactical scenarios. It illustrates that these approaches are adaptations of approaches used in wired (often fixed) infrastructures where bandwidth is known and interference is not the norm. It documents and provides experimental evidence for the Adaptive QoS (AQoS) approach that allows applications to adapt bandwidth demand to conditions without the need to know, estimate, or predict available bandwidth. AQoS informs applications that oversubscription is occurring, thereby allowing them to continue to operate, albeit at diminished rate or capacity, and meet mission needs.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Quality of service, QoS, mobile ad hoc wireless networks	79

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL